SCHÄTZUNG VON INPUT-OUTPUT TABELLEN MIT HILFE DER PSEUDOINVERSEN EINER MATRIX

Inaugural-Dissertation
zur
Erlangung des Grades eines Doktors
der
Wirtschafts- und Gesellschaftswissenschaften
durch die
Rechts- und Staatswissenschaftliche Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität
zu
BONN

vorgelegt von

JOACHIM BRAUN

aus

DESSAU

BONN 1976

SCHÄTZUNG
VON INPUT-OUTPUT TABELLEN MIT HILFE DER PSEUDOINVERSEN EINER MATRIX

Inaugural-Dissertation
zur
Erlangung des Grades eines Doktors
der
Wirtschafts- und Gesellschaftswissenschaften
durch die
Rechts- und Staatswissenschaftliche Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität
zu
BONN

vorgelegt von

JOACHIM BRAUN

aus

DESSAU

BONN 1976

Tag der Promotion: 20. Juli 1976

UNIV.
LEIDEN
BIBL.

Dekan: Professor Dr. W. Gerhardt

Erstreferent: Professor Dr. B. Korte
Zweitreferent: Professor Dr. W. Krelle

Tag der mündlichen Prüfung: 20. Juli 1976

Druck: Rheinische Friedrich-Wilhelms-Universität Bonn

Inhaltsverzeichnis

	Seite
1. Einleitung. Das Nichtnegative Matrixproblem	4
2. Spezielle Formen des Nichtnegativen Matrixproblems	8
2.1 Minimum-Norm-Verfahren	8
2.2 Biproportionale Methoden	14
2.3 Kriterien aus der Informationstheorie	17
3. Die Pseudoinverse einer Matrix	27
4. Formulierung des Problems als quadratisches Programm	33
5. Die Pseudoinverse der Transportmatrix	39
6. Berücksichtigung von Vorgabewerten	52
7. Numerische Ergebnisse	60
Anhang: Fortran-Programme	65
Literaturverzeichnis	84

1. Einleitung. Das Nichtnegative Matrixproblem

In der hier vorliegenden Arbeit betrachten wir ein in Kapitel 2.1 beschriebenes quadratisches Optimierungsproblem, das seine Anwendung beispielsweise bei der Aufstellung von Transaktionsmatrizen findet. Zur Erstellung solcher Matrizen wird ein vorgegebener Wirtschaftsraum für eine bestimmte Periode in Sektoren eingeteilt. Die Eintragungen in der Matrix geben dann sämtliche Waren- und Dienstleistungsströme wieder, die in dieser Periode fließen. Dabei ergibt sich zunächst die Frage, auf welche Weise der betrachtete Wirtschaftsraum in Sektoren aufgeteilt werden soll. In Stäglin [25] sind die zum Beispiel bei einer gesamtvolkswirtschaftlichen Tabelle zu berücksichtigenden Prinzipien dargestellt. Wir gehen im folgenden davon aus, daß diese Einleitung bereits geschehen ist.

Ein weiteres Problem bildet die Materialbeschaffung. Im Zusammenhang damit können folgende beispielsweise in [13] beschriebene Situationen auftreten:

(a) Eine für einen bestimmten Zeitpunkt vorliegende Tabelle soll dadurch aktualisiert werden, daß sie in eine Matrix mit bekannten Zeilen- und Spaltensummen, die von denen der alten Matrix abweichen, umgeformt wird. Es liegt hier also das Problem der Prognose bestimmter Werte vor.

(b) Es steht genügend statistisches Material zum Ausfüllen der gesamten für die betrachtete Periode zu bestimmenden Tabelle vor. Im allgemeinen entstammen diese Daten jedoch verschiedenen, voneinander unabhängigen Quellen. Deshalb kann es vorkommen, daß einige Quellen Information über die Matrixelemente selbst liefern, während andere Quellen unabhängig davon bestimmte Zeilen- und Spaltensummen festlegen, die mit den Zeilen- und Spaltensummen, die man aus der Matrix erhält, nicht übereinstimmen. Falls man weiß oder annimmt, daß die neuen Randsummen gesicherter sind, ergibt sich also die Frage, wie man die Matrixelemente fest vorgegebenen Zeilen- und Spaltensummen anpassen soll.

(c) Es liegt dieselbe Situation wie in (b) vor mit dem Zusatz, daß nicht alle Matrixelemente geändert werden dürfen, sondern daß einige davon feste Werte annehmen sollen. Man hat hier also im Vergleich zu (b) weniger Freiheitsgrade für die Bestimmung der Tabellenelemente.

Diese Beispiele skizzieren die Gründe, deretwegen man dazu übergegangen ist, Modelle zur Erstellung von Transaktionsmatrizen zu konstruieren: Einmal zur Prognose bestimmter Ereignisse, zum anderen wegen der Schwierigkeiten, die bei der Materialbeschaffung auftreten. Die einzelnen Modelle entscheiden sich in dem Kriterium, nach dem die Matrixelemente an die neuen Zeilen- und Spaltensummen angepaßt werden. Für welches Kriterium man sich dabei entscheidet, hängt von der jeweiligen konkreten Situation ab. Wenn ein Modell in einem Zusammenhang inhaltlich sinnvoll begründet werden kann, so braucht dies in einem anderen Zusammenhang nicht notwendig der Fall zu sein. So ist beispielsweise der Problemursprung im obigen Fall (a) anders als bei (b) und (c).

Unabhängig davon führt die Modellbildung zu folgendem formalen Problem, das wir in Anlehnung an Bacharach [4] als "Nichtnegatives Matrixproblem mit Nebenbedingungen" oder einfach als "Nichtnegatives Matrixproblem" bezeichnen:

Gegeben seien eine reelle(m,n)-Matrix

$A=((a_{ij}))$, Zahlen $p_i>o$, $i=1,\ldots,m$; $q_j>o$, $j=1,\ldots,n$,

mit

$$\sum_{i=1}^{m} p_i = \sum_{j=1}^{n} q_j \ ,$$

eine (möglicherweise leere) Teilmenge R von $\{1,\ldots,m\}x\{1,\ldots,n\}$ und Zahlen $c_{ij} \geq o$ für jedes $(i,j)\varepsilon R$.

Gesucht ist eine (m,n)-Matrix $X=((x_{ij}))$, die folgende Bedingungen erfüllt:

$$\text{(i)} \qquad \sum_{j=1}^{n} x_{ij} = p_i \qquad \forall i$$

$$\text{(ii)} \qquad \sum_{i=1}^{m} x_{ij} = q_j \qquad \forall j$$

$$\text{(iii)} \qquad x_{ij} \geq 0 \qquad \forall i,j$$

$$\text{(IV)} \qquad x_{ij} = c_{ij} \qquad \forall (i,j) \in R$$

(V) die Elemente von X erfüllen ein weiteres, fest vorgegebenes Kriterium.

Die aus der Literatur bekannten Varianten von (V) lassen sich folgendermaßen klassifizieren:

1. Minimum-Norm-Verfahren
2. Biproportionale Methoden
3. Kriterien aus der Informationstheorie

Da Modelle zur Überwindung statistisch-empirischer Schwierigkeiten bzw. zur Prognose zukünftiger Ereignisse gebildet werden, ist es eine wesentliche Frage, inwieweit ein bestimmtes Modell einer numerischen Behandlung zugänglich ist. Bei Punkt (V) ist in vielen Fällen die Forderung naheliegend, daß die gesuchte Matrix "nicht allzuweit" von der Ausgangsmatrix "entfernt" ist. Dies führt in natürlicher Weise auf ein Normminimierungsproblem, jedoch haben die bisher vorliegenden Ansätze, Modelle der ersten Problemklasse zu behandeln, in der Praxis im wesentlichen versagt (siehe dazu Kapitel 2.1). Die in dieser Arbeit behandelte Version von (V) gehört ebenfalls zur Kategorie der Minimum-Norm-Verfahren und ist das Ergebnis des Versuchs, diese Schwierigkeiten zu überwinden. Dabei machen wir einen Ansatz unter Benutzung der Pseudoin-

versen von Matrizen.

Die Arbeit ist folgendermaßen gegliedert:

Im anschließenden Kapitel wird das hier gewählte Kriterium im Zusammenhang mit anderen Ansätzen beschrieben. In Kapitel 3 werden die von uns benötigten Grundtatsachen über die Pseudoinverse bereitgestellt, Kapitel 4 enthält die Formulierung des Problems als duales quadratisches Programm mit Hilfe der Pseudoinversen der hier vorliegenden Restriktionenmatrix, in Kapitel 5 wird diese Pseudoinverse explizit berechnet. Die angegebene Darstellung legt eine Lösung des dualen Programms mittels geeigneter Verfahren der zyklischen Minimierung nahe, da dafür die Iterationsvektoren eine besonders einfache Gestalt annehmen. Diese Vektoren werden ebenfalls angegeben. In Kapitel 6 wird das Problem unter Berücksichtigung von Vorgabewerten behandelt (Forderung (IV) oben), Kapitel 7 enthält numerische Ergebnisse. Als Anhang sind Fortran-Programme zur Lösung des Problems mit bzw. ohne Vorgabewerte beigefügt.

2. Spezielle Formen des Nichtnegativen Matrixproblems

2.1 Minimum-Norm-Verfahren

In Stäglin [25] findet man folgenden Ansatz:

Sei

$$u_i := \sum_{j=1}^{n} a_{ij} \,, \quad v_j := \sum_{i=1}^{m} a_{ij} \,.$$

Dann wird eine erste Näherung $\bar{A}$ an die gesuchte Matrix X definiert durch

$$\bar{a}_{ij} := \sqrt{\frac{p_i}{u_i} \cdot \frac{q_j}{v_j}} \cdot a_{ij} \,.$$

$\bar{A}$ erfüllt im allgemeinen nicht die neuen Zeilen- und Spaltensummen, deshalb wird die Funktion

$$f_1 := \sum_{i=1}^{m} \sum_{j=1}^{n} (x_{ij} - \bar{a}_{ij})^2 \qquad (1)$$

unter den Nebenbedingungen (i) bis (IV) minimiert.
Matuszewski, Pitts und Sawyer [16] betrachten

$$f_2 := \sum_{i=1}^{m} \sum_{j=1}^{n} |x_{ij} - a_{ij}| \qquad (2)$$

und schätzen damit die kanadische Input-Output-Tabelle.

Die Funktionen (1) und (2) induzieren Normen. Friedlander [10] wählt einen Ansatz über Halbnormen, indem er $a_{ij} \geq o$ voraussetzt und die Funktion

$$f_3 := \sum_{(i,j)\in N} \frac{(x_{ij}-a_{ij})^2}{a_{ij}} \qquad (3)$$

betrachtet, wobei $N := \{(i,j) \mid a_{ij} > o\}$.

Die Frage nach der numerischen Lösbarkeit kann positiv beantwortet werden im Falle der in [16] betrachteten Funktion (2). Daß dieser Ansatz in der Praxis durchführbar war, liegt daran, daß das Problem

$$\min \sum_{i,j} |x_{ij} - a_{ij}| \quad \text{unter (i) bis (iv)}$$

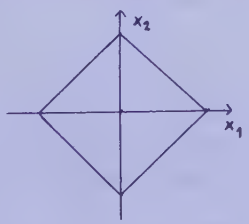
durch Hinzufügen geeigneter Nebenbedingungen in ein äquivalentes lineares Programm umgeformt werden kann, und bekanntlich existieren leistungsfähige Algorithmen zur Lösung auch großer linearer Programme.

Trotz dieses Vorteils erscheinen jedoch Bedenken gegen die Wahl von f_2 angebracht. Die Funktion (2) ist nicht streng konvex, hat also im allgemeinen keinen eindeutig bestimmten Minimanden, und wenn mehr als ein optimaler Punkt existiert, so auch gleich überabzählbar viele. Man braucht also ein zusätzliches Kriterium, um zu entscheiden, welchen Minimanden man wählen soll. Die folgende Überlegung macht die dabei auftretenden Schwierigkeiten deutlich:

Im $\mathbb{R}^k$ ist die Normkugel mit Radius r um einen festen Punkt $a \in \mathbb{R}^k$ bezüglich der durch f_2 induzierten Norm

$$\|x\|_2 := \sum_{i=1}^{k} |x_i|$$

der Durchschnitt von 2^k Halbräumen. Sie hat beispielsweise für $k=2$, $r=1$, $a=o$ folgende Gestalt:



Die Nebenbedingungen (i), (ii), (iii) definieren ein Transportpolyeder T im $\mathbb{R}^{m \cdot n}$. Bekanntlich hat T die Dimension $(m-1)\cdot(n-1)$, siehe Klee, Witzgall [14]. Deshalb kann es vorkommen, daß T für ein passendes $r>0$ ganz in einer der Hyperebenen liegt, die die Sphäre mit Radius r um den Punkt a mit den $m \cdot n$ Matrixkomponenten a_{ij} begrenzen. Das folgende Beispiel zeigt einen solchen Fall:

Sei $m=n=2$, $p_1=p_2=q_1=q_2=\frac{1}{2}$, sei

$$A := \begin{pmatrix} \frac{1}{2} & \frac{3}{2} \\ 1 & 1 \end{pmatrix}$$

Dann hat jede bezüglich der Nebenbedingungen (i), (ii), (iii) zulässige Matrix X die Form

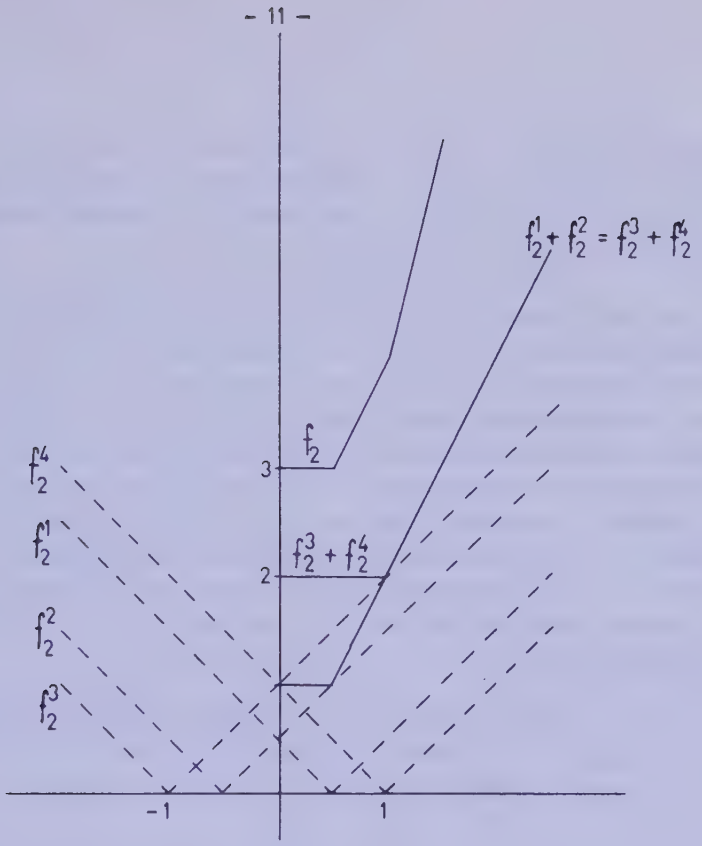
$$X = \begin{pmatrix} x & -x+\frac{1}{2} \\ -x+\frac{1}{2} & x \end{pmatrix}, \quad 0 \leq x \leq \frac{1}{2} .$$

Sei nun

$$f^1{}_2 := |x_{11}-a_{11}| = |x-\tfrac{1}{2}|, \quad f^2{}_2 := |x_{21}-a_{21}| = |-x-\tfrac{1}{2}|,$$

$$f^3{}_2 := |x_{12}-a_{12}| = |-x-1|, \quad f^4{}_2 := |x_{22}-a_{22}| = |x-1|.$$

Dann ist f_2 im gesamten zulässigen Bereich konstant mit $f_2(x)=3$ für alle x. Das folgende Bild mag zur Veranschaulichur dienen:



Bei Wahl der Funktion f_2 kann also der Fall eintreten, daß sogar jeder zulässige Punkt optimal ist. Es ist klar, daß ein solches Modell keine große Aussagekraft hat. Für größere Werte von m und n hat man im allgemeinen keine Kontrolle mehr darüber, wie das Polyeder T und die Sphären $f_2(x)$ = konstant zueinander liegen. Man kann also nicht entscheiden, ob es sinnvoll ist, ein Modell unter Zuhilfenahme der Funktion f_2 zu konstruieren.

Falls ein Indexpaar (i,j) mit $a_{ij}=o$ existiert, so ist die Funktion f_3 ebenfalls nicht streng konvex. Aus allgemeinen Sätzen über die Optimalpunkte von konvexen quadratischen Programmen folgt jedoch, daß die Optimalpunkte von f_3 unter den Nebenbedingungen (i) bis (iii) wenigstens in den Komponenten (i,j) mit $a_{ij}>o$ eindeutig bestimmt sind. Man erhält also

nur einen optimalen Punkt, falls man fordert, daß Nullelemente in der Ausgangsmatrix in der Lösungsmatrix erhalten bleiben, oder (was bei dem obigen Beispiel einer (2,2)-Matrix der Fall ist) falls alle a_{ij} positiv sind.

Diese Bemerkungen zeigen, daß aus theoretischer Sicht die oben angeführten quadratischen Funktionen gegenüber der Funktion f_2 gewisse Vorteile haben, jedoch waren Versuche, Probleme dieser Form auf einer elektronischen Rechenanlage mit gängigen Methoden der quadratischen Optimierung zu lösen, aus Speicherplatz- und Zeitgründen höchstens für kleine Matrizen erfolgreich. So bemerkt Stäglin [25] , daß der Ansatz (1) vom Deutschen Institut für Wirtschaftsforschung aufgegeben wurde, da zu seiner exakten Lösung für große Matrizen ein erheblicher Rechenaufwand erforderlich war. In Bacharach [4] wird diese Erfahrung bestätigt.

In der hier vorliegenden Arbeit behandeln wir das Nichtnegative Matrixproblem als Normierungsproblem bezüglich der euklidischen Norm, genauer betrachten wir die Funktion

$$f := \sum_{i=1}^{m} \sum_{j=1}^{n} (x_{ij} - a_{ij})^2 \qquad (4)$$

und minimieren (4) unter den Bedingungen (i) bis (iv). Unter Benutzung der Pseudoinversen der das Gleichungssystem (i), (ii), (iv) bestimmenden Matrix und Übergang zum dualen Problem (siehe Kapitel 4 und 6) können dann Aufgaben in praxisrelevanten Größenordnungen von m und n durch zyklische Minimierung gelöst werden. Der dabei benötigte Speicherplatz ist wesentlich geringer als bei anderen bekannten Verfahren der quadratischen Optimierung (siehe dazu Kapitel 7). Man beachte, daß auf Grund der formalen Gleichheit der Funktionen (4) und (1) damit auch der Ansatz in [25] behandelt werden kann.

Die Anwendung der Pseudoinversen in der Mathematischen Programmierung ist an sich nichts Neues. Da die Lösungsmannigfaltigkeit von linearen Gleichungssystemen mit Hilfe der Pseudoinversen beschrieben werden kann (siehe Kapitel 3), liegt es nahe, die Pseudoinverse insbesondere zur Lösung von linearen oder quadratischen Programmen unter linearen Nebenbedingungen heranzuziehen. Solche Ansätze sind in der Tat gemacht worden, jedoch ist die zum Beispiel in [17] vorgeschlagene primale Methode zur Lösung eines quadratischen Programms höchstens für kleine Probleme durchführbar, und auch Pyles Gradientenprojektionsverfahren [19], [20] für lineare Programme bietet nicht unerhebliche Schwierigkeiten. Einer der neuesten Ansätze dieser Art ist die Arbeit von Shankland [21]. Die dort vorgeschlagene Methode hat jedoch zwei wesentliche Nachteile: ihre Anwendung auf das Nichtnegative Matrixproblem erfordert erstens einen beachtlichen Speicheraufwand, zum anderen liegt bisher kein Konvergenzbeweis vor.

2.2 Biproportionale Methoden

Eine andere Variante von (V) ist die Forderung, daß die gesuchte Matrix X biproportional zu A ist, wobei folgende Definition zugrundegelegt wird:

Definition Seien A,B nichtnegative (m,n)-Matrizen.
B heißt echt biproportional zu A: $\Longleftrightarrow$ Es gibt Diagonalmatrizen R und S, so daß

$$B = R A S \tag{5}$$

Sei $\mathfrak{L}(A)$ die Menge aller zu A echt biproportionalen Matrizen.
B heißt biproportional zu A: $\Longleftrightarrow$ Es gibt eine Folge $\langle B^t \mid t \in \mathbb{N} \rangle$, so daß

$$B^t \in \mathfrak{L}(A) \quad \text{für alle} \quad t$$

und

$$B = \lim_{t \to \infty} B^t \tag{6}$$

Wir bezeichnen die Menge aller zu A biproportionalen Matrizen mit $\mathfrak{L}^*(A)$. Man beachte, daß Biproportionalität nur für nichtnegative Matrizen definiert ist. (Die Matrix B^t bezeichnet natürlich das t-te Glied der Folge und nicht das t-fache Produkt von B).

Es gibt zwei Methoden, um zu einer gegebenen Matrix A ein $B \in \mathfrak{L}^*(A)$ zu finden: das RAS-Verfahren [27] , das seinen Namen wegen der Darstellung (5) erhalten hat, und das Modop-Verfahren [23] , [26] . Beide Verfahren konstruieren ausgehend von einer Ausgangsmatrix A^0 eine gegen die gesuchte Matrix konvergierende Folge von Matrizen, indem abwechselnd die Zeilen- und Spaltensummen korrigiert werden, wobei die beiden Verfahren sich nur in der Wahl der Ausgangsmatrix unterscheiden.

Bei RAS wird

$$A^{o} := A$$

gewählt, Modop beginnt mit der durch

$$a^{o}_{ij} := \sqrt{\frac{p_i}{u_i} \cdot \frac{q_j}{v_j}} \cdot a_{ij}$$

definierten Matrix A^{o}, wobei

$$u_i := \sum_{j=1}^{n} a_{ij}, \quad v_j := \sum_{i=1}^{m} a_{ij}.$$

Der allgemeine Iterationsschritt hat dann folgende Form: Sei A^t schon bekannt, t=o,1,2,.... Falls t=2k gerade: (k=o,1,2,...).

$$\text{Sei} \quad r_i^k := \frac{p_i}{\sum_{j=1}^{n} a_{ij}^t} \qquad \forall i$$

$$R^k := \operatorname{diag}\left[r_1^k, \ldots, r_m^k\right]$$

$$A^{t+1} := R^k \cdot A^t.$$

Falls t=2k+1 ungerade:

$$\text{Sei} \quad s_j^k := \frac{q_j}{\sum_{i=1}^{m} a_{ij}^t} \qquad \forall j$$

$$S^k := \operatorname{diag}\ s_1^k, \ldots, s_n^k$$

$$A^{t+1} := A^t \cdot S^k.$$

Biproportionale Lösungen erfüllen die schon in Kapitel 2.1 erwähnte Forderung der Eindeutigkeit. Genauer gilt folgender Satz, den wir nur für das RAS-Verfahren formulieren, der aber genauso für Modop gilt:

Satz (Bacharach [4])

(a) Wenn $B=((b_{ij}))$ eine RAS-Lösung des biproportionalen Problems unter den Nebenbedingungen (i), (ii) ist, so auch unter den Nebenbedingungen (i), (ii), (iii) und (vi), wobei

$$(vi) \qquad (i,j)(a_{ij}=o \Longrightarrow b_{ij}=o).$$

(b) Wenn überhaupt ein X existiert, das (i), (ii), (iii) und (vi) erfüllt, so auch ein $B \in \mathcal{L}^*(A)$. Dieses B ist eindeutig bestimmt und wird durch das RAS-Verfahren gefunden.

Falls zusätzlich

$$(iv) \qquad x_{ij} = c_{ij}$$

mit $c_{ij}>o$ für eine Menge $R \subset \{1,\ldots,m\}x\{1,\ldots,n\}$ erfüllt sein soll, so fordert man vernünftigerweise Biproportionalität nur für die nicht in R liegenden Indexpaare. Wegen des obigen Satzes kann das dadurch erreicht werden, daß man die Vorgabewerte c_{ij} von den neuen Zeilen- und Spaltensummen subtrahiert und die entsprechenden Komponenten der Ausgangsmatrix Null setzt.

Der biproportionale Ansatz ist seiner Natur nach kein Optimerungsansatz. Es gibt jedoch eine aus der Informationstheorie stammende Funktion, deren Minimand gerade die biproportionale Lösung des Nichtnegativen Matrixproblems ist. Im Hinblick darauf kann die Lösungsmatrix in einer gewissen Weise interpretiert werden. Diese und damit zusammenhängende Fragen werden im folgenden Abschnitt behandelt.

2.3 Kriterien aus der Informationstheorie

Theil [28] bzw. Bacharach [4] minimieren die unten angegebenen aus der Informationstheorie stammenden Funktionen (12) bzw. (9) unter den Nebenbedingungen (i) bis (iv), wobei die von Bacharach betrachtete Minimumaufgabe (9) zu einer biproportionalen Lösung führt. Die Bedeutung der hier betrachteten Funktionen ist im Gegensatz zu den Minimum-Norm-Kriterien aus Kapitel 2.1 möglicherweise nicht unmittelbar klar. Deshalb sollen einige Grundtatsachen aus der Informationstheorie vorausgeschickt werden. Dabei wollen wir vorwiegend den intuitiven Aspekt betonen und einen zu starken Einstieg in den mathematischen Formalismus vermeiden, da ein systematischer Aufbau der Informationstheorie natürlich nicht in den Rahmen dieser Arbeit gehört, und in dem hier vorliegenden Zusammenhang vor allem eine inhaltliche Interpretation der Ansätze in [28] und [4] im Hinblick auf das Nichtnegative Matrixproblem interessiert. Bezüglich einer formalen (axiomatischen) Begründung der Informationstheorie sei auf die Literatur verwiesen, siehe zum Beispiel [3] , [24] , [29] .

Die Informationstheorie in ihrer heutigen Form geht auf Shannon [22] zurück und geht von folgendem Problem aus:

Sei E ein Experiment. Bevor es ausgeführt wird, herrscht "Unsicherheit" oder "Ungewissheit" über seinen Ausgang. Die Frage ist nun, wie man diesen umgangssprachlichen Begriff präzisieren und damit einer mathematischen Behandlung zugänglich machen kann. Statt Unsicherheit verwendet man häufig den aus der Physik stammenden Begriff Entropie.

Wir betrachten im folgenden nur Versuche E mit endlich vielen sich gegenseitig ausschließenden möglichen Ausgängen $E_1,\ldots,E_n$. Sei x_i die a priori vorgegebene Wahrscheinlichkeit, mit der das Ergebnis E_i eintritt, d.h. $x_i=P(E_i)$, sei $X = (x_1,\ldots,x_n)$. Dann kann das obige Problem auch in folgender Form gestellt werden: Vor Beginn des Experiments weiß man

nichts über seinen Ausgang. Wenn das Versuchsergebnis E_i bekannt ist, ist man in Abhängigkeit von x_i mehr oder weniger überrascht, wobei die Überraschung darüber, daß E_i tatsächlich beobachtet worden ist, desto größer sein wird, je unwahrscheinlicher das Eintreten von E_i a priori war. Wir stellen uns nun vor, daß der Versuch beliebig oft durchgeführt wird, wobei je zwei Ausführungen von E unabhängig voneinander geschehen mögen, und notieren jedesmal den jeweiligen Versuchsausgang. Gesucht wird dann eine Funktion H(X), die den Versuch dadurch charakterisiert, daß sie ein vernünftiges Maß dafür angibt, wie groß die im Durchschnitt oder Mittel pro Einzelversuch zu erwartende "Überraschung" über den Versuchsausgang ist. Dabei sollte die Funktion, die die Überraschung über das Eintreten eines einzelnen Elementarereignisses E_i beschreibt, in Abhängigkeit von x_i streng monoton fallen. Seit Shannon hat sich dafür der Ausdruck

$$\log \frac{1}{x_i}$$

durchgesetzt. Demgemäß definiert man als Mittelwert die Größe

$$H(X) := \sum_{i=1}^{n} x_i \log \frac{1}{x_i} = - \sum_{i=1}^{n} x_i \log x_i$$

und nennt H(X) die Entropie von X. Die Wahl der Basis ist dabei unwesentlich, da für je zwei verschiedene Basen die entsprechenden Logarithmusfunktionen proportional sind. Häufig wird die Basis 2 gewählt. Wir legen im folgenden die obige Definition der Entropie zugrunde. Daneben gibt es eine Reihe weiterer, weniger bekannter Charakterisierungen, siehe zum Beispiel [1].

Die folgende Interpretation ist für unsere Zwecke besonders nützlich. Wir stellen uns wieder vor, der Versuch würde unendlich oft ausgeführt und betrachten zwei Personen P_1 und P_2, wobei P_1 jedesmal den Versuchsausgang beobachten kann, P_2 aber nicht. P_2 versucht nun, durch Fragen an P_1 das Versuchsergebnis festzustellen, wobei P_1 nur mit "Ja" oder "Nein" antworten darf. Person P_2 kann dabei natürlich eine Menge überflüssiger Fragen stellen. Gesucht ist also eine Fragestrategie S, die optimal ist in dem Sinne, daß möglichst wenig Ja und Nein Antworten notwendig sind, um das Ergebnis zu erfahren. Dann kann die Entropie folgendermaßen interpretiert werden, siehe zum Beispiel Topsóe [29] :

(a) Für eine beliebige Fragestrategie S sei f_i die Anzahl der Fragen, die notwendig sind, um den Ausgang E_i festzulegen. Dann ist

$$\sum_{i=1}^{n} x_i f_i$$

die Anzahl der Fragen, die man bei Benutzung von S im Durchschnitt pro Einzelversuch stellen muß, um den Versuchsausgang zu erfahren.

(b) H(X) ist die durchschnittliche Anzahl der Fragen pro Einzelversuch, die man stellen muß, um den Ausgang zu erfahren, wenn die Fragen optimal gestellt werden. Dabei gibt die Größe

$$\log \frac{1}{x_i}$$

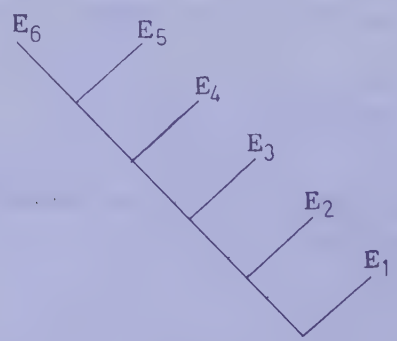
die durchschnittliche Zahl der Fragen an, die notwendig sind, um den i-ten Ausgang festzulegen.

Diese beiden Aussagen sollen kurz an einem Beispiel erläutert werden. Da wir uns mit "Ja" und "Nein" Antworten befassen, ist es nützlich, sich zu einer gegebenen Fragestrategie S einen binären Wurzelbaum zu konstruieren, wobei die Endknoten die möglichen Ereignisse symbolisieren. Lautet die Antwort auf eine Frage "Nein", so wird, indem man mit der ersten Frage bei der Wurzel beginnt, nach links verzweigt, anderenfalls nach rechts. Die Anzahl der Fragen, die benötigt werden, um ein Ereignis E_i festzulegen, ist damit gleich der Anzahl der Kanten von der Wurzel bis zu dem Endknoten, der E_i repräsentiert.

Wir betrachten nun ein Experiment mit sechs möglichen Ausgängen, deren Wahrscheinlichkeiten der folgenden Tabelle zu entnehmen sind:

Ereignis	E_1	E_2	E_3	E_4	E_5	E_6
Wahrscheinlichkeit	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$

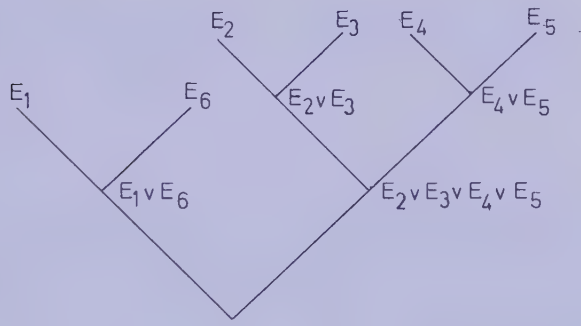
Eine mögliche Strategie besteht darin, daß die Ereignisse in aufsteigender Indexfolge erfragt werden, d.h. zuerst fragt man "Tritt E_1 ein?", danach "Tritt E_2 ein?" u.s.w. Dies führt zu folgendem Baum:



Die durchschnittliche Anzahl der Fragen ergibt sich also zu

$$\frac{1}{4} \cdot 1 + \frac{1}{8} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 4 + \frac{1}{8} \cdot 5 + \frac{1}{4} \cdot 5 = 3{,}25.$$

Eine andere Strategie wird durch den folgenden Baum dargestellt:



In diesem Fall ist die durchschnittliche Anzahl der Fragen

$$\frac{1}{4} \cdot 2 + \frac{1}{4} \cdot 2 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 + \frac{1}{8} \cdot 3 = 2{,}5.$$

Legt man die Basis 2 für den Logarithmus zugrunde, so erkennt man, daß die zweite Strategie optimal ist, denn mit den obigen Bezeichnungen ergibt sich

$$f_i = \log \frac{1}{x_i} \quad \text{und} \quad H(X) = 2{,}5.$$

Um zu den bei Theil bzw. Bacharach angeführten Funktionen zu gelangen, betrachten wir nun den Fall, daß sich die a priori-Wahrscheinlichkeiten $X=(x_1,\ldots,x_n)$ ändern zu Wahr-

scheinlichkeiten $X^* = (x_1^*, \ldots, x_n^*)$, wobei eine Person P_1 diese Veränderung bemerkt, eine zweite Person P_2 aber nicht. Ursprünglich hatten beide Personen ihre Fragestrategie optimal den Wahrscheinlichkeiten x_i angepaßt. Da Person P_1 den Übergang von X zu X^* bemerkt, ändert sie ihre Strategie sofort und braucht

$$- \sum_{i=1}^{n} x_i^* \cdot \log x_i^*$$

Fragen, um den Ausgang zu erfahren. P_2 bemerkt die Änderung nicht, arbeitet weiterhin mit der alten Strategie und braucht nach (a)

$$- \sum_{i=1}^{n} x_i^* \cdot \log x_i$$

Fragen. Dann muß P_2 also

$$-\Sigma\, x_i^* \cdot \log x_i - \left[\Sigma - x_i^* \cdot \log x_i^*\right] = \Sigma\, x_i^* \cdot \log \frac{x_i^*}{x_i} \qquad (7)$$

Fragen mehr als P_1 stellen. Man kann zeigen, daß der Ausdruck (7) nie negativ ist. Diese Überlegung führt zu folgender Definition:

Die Funktion

$$I\,(X^* | X) := \sum_{i=1}^{n} x_i^* \cdot \log \frac{x_i^*}{x_i}$$

heißt Informationsdivergenz zwischen X^* und X. Die Bezeichnung ist dabei in der Literatur nicht einheitlich.

So wird statt Informationsdivergenz häufig der Begriff Informationsgewinn verwendet.

Wenn wir wieder zu der ursprünglichen Auffassung der Entropie H(X) als "Überraschung" zurückkehren, so kann die Größe $I(X^*||X)$ folgendermaßen interpretiert werden: Sie gibt die Überraschung an, die eine von den Werten x_i ausgehende Person P_2 zusätzlich zu der Überraschung erlebt, die eine Person P_1 erlebt, die die wahren Werte x_i^* kennt.

Man beachte, daß im allgemeinen

$$I(X^*||X) \neq (X||X^*) \tag{8}$$

gilt.

Die Informationsdivergenz wird nun in [4] und [28] zur Lösung des Nichtnegativen Matrixproblems benutzt, wobei sich die Ansätze im wesentlichen dadurch unterscheiden, daß Bacharach eine Funktion der Form $I(X^*||X)$ betrachtet, Theil hingegen $I(X||X^*)$. Wegen der Ungleichung (8) erhält man im allgemeinen auch verschiedene optimale Punkte. Als Ausgangspunkt dient folgende Überlegung: Jedes Element y_{ij} der Matrix ist die absolute Größe (oder die absolute Häufigkeit) einer Transaktion von i nach j, also ist

$$(*) \qquad \bar{y}_{ij} := \frac{y_{ij}}{S} \quad \text{mit} \quad S := \sum_{i=1}^{m} \sum_{j=1}^{n} y_{ij}$$

der relative Anteil aller Transaktionen (oder die relative Häufigkeit einer Transaktion von i nach j). Im folgenden denken wir uns sowohl die Ausgangsmatrix $A = ((a_{ij}))$ mit Zeilensummen

$$u_i = \sum_{j=1}^{n} a_{ij}$$

und Spaltensummen

$$v_j = \sum_{i=1}^{m} a_{ij}$$

als auch die gesuchte Matrix $X = ((x_{ij}))$ mit Zeilensummen p_i und Spaltensumme q_j gemäß (*) normiert und lassen die Querstriche weg. In [4] wird dann die Funktion

$$I(X||A) = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \log \frac{x_{ij}}{a_{ij}} \qquad (9)$$

unter den Nebenbedingungen (i) bis (iv) aus Kapitel 1 minimiert.

Theil setzt zunächst

$$d_{ij} := \frac{p_i}{u_i} \cdot \frac{q_j}{v_j} \cdot a_{ij} \qquad (10)$$

$$b_{ij} := \frac{d_{ij}}{\sum_{k,l} d_{kl}} \qquad (11)$$

und minimiert dann

$$I(B||X) = \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij} \log \frac{b_{ij}}{x_{ij}} \qquad (12)$$

Man beachte, daß in beiden Fällen die x_{ij} die Variablen sind.

Der Zusammenhang zwischen dem (wegen der strengen Konvexität) eindeutig bestimmten Minimanden von (9) und biproportionalen Matrizen wird durch folgenden Satz ausgedrückt. Der Beweis folgt sofort aus einem Lagrange-Ansatz:

Satz Falls eine zu A echt biproportionale Matrix X existiert, die (i) und (ii) erfüllt, so ist X die eindeutig bestimmte Lösung des Problems

$$\min I(X||A) \quad \text{unter (i), (ii) und (iii).}$$

Im Hinblick auf die obigen Bemerkungen über $I(X||A)$ kann der Bacharach'sche Ansatz folgendermaßen interpretiert werden:

Man betrachtet die Matrix A als Schätzung oder Voraussage für Wahrscheinlichkeiten, mit denen bestimmte Ereignisse eintreten werden. Diese Schätzung ist offenbar noch mit Fehlern behaftet, da A die neuen Zeilen- und Spaltensummen nicht erfüllt. Da man die "wirklichen" Werte nicht kennt, bestimmt man, ausgehend von der Vorhersage A, eine Matrix X von Wahrscheinlichkeiten derart, daß die Überraschung, die man erlebt, wenn die Ereignisse mit eben diesen Wahrscheinlichkeiten x_{ij} einträten, möglichst klein wird.

Theils Ansatz (12) kann dann aufgefaßt werden als der Betrag an "neuer Überraschung", den die Nachricht enthält, daß b_{ij} eingetreten ist, wenn x_{ij} vorausgesagt wurde.

Man beachte den konzeptionellen Unterschied zwischen (9) und (12): Bei (9) geht man von einer festen Voraussage aus und sucht eine mögliche Realisierung, während beim Ansatz (12) gewissermaßen der umgekehrte Weg beschritten wird,

indem ausgehend von einer Realisierung eine Vorhersage bestimmt wird.

In der Einleitung wurde die Forderung erwähnt, daß die gesuchte Matrix "möglichst nahe" an der Ausgangsmatrix liegen soll. Die bisherigen Ausführungen zeigen, daß auch die beschriebenen informationstheoretischen Ansätze als Erfüllung dieser umgangssprachlichen Formulierung interpretiert werden können. Die durch (9) bzw. (12) bestimmten Matrizen liegen in dem oben jeweils beschriebenen Sinne "in der Nähe" der Ausgangsmatrix. Ebenso wie bei den Normminimierungskriterien hat man auch hier verschiedene Funktionen zur Auswahl, wobei einer der möglichen Ansätze eine biproportionale Lösung liefert.

3. Die Pseudoinverse einer Matrix

Dieses Kapitel enthält die Definition der Pseudoinversen einer Matrix und eine kurze Zusammenfassung ihrer grundlegenden Eigenschaften. Für Beweise sei auf die Literatur verwiesen, z. B. [2] , [5] , [6] und [12] . [5] und [12] enthalten auch weitergehende Ergebnisse über die Pseudoinverse von linearen Abbildungen zwischen nicht notwendig endlichdimensionalen Räumen. Wir wollen zunächst einige Bezeichnungen und Begriffe vorausschicken, von denen wir einen Teil schon stillschweigend in Kapitel 2 benutzt haben, die aber hier noch einmal systematisch zusammengestellt werden sollen.

3.1 Im folgenden bedeutet Vektorraum immer reeller Vektorraum.

3.2 Für jede (m,n)-Matrix A bezeichnet A^T die zu A transponierte Matrix. Dies gilt insbesondere für Vektoren.

3.3 Seien X, Y Vektorräume, sei $\alpha: X \to Y$ linear. Dann heißt

$$N(\alpha) := \{x \in X \mid \alpha x = o\}$$

der Kern von α und

$$R(\alpha) := \{y \in Y \mid \exists x \in X: \alpha x = y\}$$

das Bild von α.

3.4 Sei X ein Vektorraum, f ein Skalarprodukt für X.
(i) Seien $x, y \in X$. x heißt orthogonal zu y bezüglich f (Bezeichnung: $x \perp y$): $\Longleftrightarrow f(x,y)=o$.

(ii) Für jede Teilmenge U von X heißt

$$U^{\perp} := \{y \in X \mid y \perp x \ \forall\, x \in X\}$$

das orthogonale Komplement von U in X.

(iii) Die durch f induzierte Norm $\|\cdot\|$ ist definiert durch

$$\|x\| := \sqrt{f(x,x)} \qquad \forall\, x \in X.$$

(iv) Wenn X endlichdimensional ist (d.h. dim X=n für ein passendes $n \in \mathbb{N}$), so kann f bezüglich einer Basis $\{b_1,\ldots,b_n\}$ von X eine positiv definite (n,n)-Matrix $Q = ((q_{ij}))$ zugeordnet werden durch

$$q_{ij} := f(b_i, b_j),$$

und f hat dann die Darstellung

$$f(x,y) = \xi^T Q \eta,$$

wobei ξ,η die Koordinatenvektoren von x,y sind.

3.5 Sei X ein Hilbertraum mit Skalarprodukt f, sei U ein abgeschlossener Vektorunterraum von X. Dann kann jedes $x \in X$ in eindeutiger Weise geschrieben werden als

$$x = y + v$$

mit $y \in U$ und $v \in U$, und die orthogonale Projektion

$$P_U : X \to U$$

ist definiert durch

$$P_U x := y .$$

3.6 Wenn X und Y endlichdimensionale Vektorräume sind, so kann bekanntlich jeder linearen Abbildung $\alpha: X \to Y$ bezüglich fester Basen von X und Y eine Matrix A zugeordnet werden. Von jetzt an betrachten wir nur noch Vektorräume $X = \mathbb{R}^n$, $Y = \mathbb{R}^m$ für passende m,n, wählen jeweils die kanonische Basis und unterscheiden deshalb nicht zwischen linearen Abbildungen und Matrizen.

3.7 Seien X bzw. Y Vektorräume mit Skalarprodukten f bzw. g, sei A: $X \to Y$ linear. Die Adjungierte A^* von A ist die Abbildung

$$A^* : Y \to X$$

definiert durch

$$f(x, A^* y) = g(Ax, y) \quad \forall x \in X, \quad \forall y \in Y.$$

Wenn R die Matrix von f und S die Matrix von g ist, so kann A^* geschrieben werden

$$A^* = R^{-1} . A^T . S.$$

Als nächstes soll die Pseudoinverse einer Matrix definiert werden. Dies kann einmal rein algebraisch geschehen, indem man die unten angegebenen Gleichungen (i) bis (iv) aus 3.9 an den Anfang stellt und die Pseudoinverse als die eindeutige Lösung dieser Gleichungen definiert. Dies erscheint jedoch etwas willkürlich, da die Motivation für diese vier Gleichungen nicht unmittelbar einleuchtet. Deshalb soll hier der zweite mögliche Weg beschritten werden, der die geometrischen Eigenschaften der Pseudoinversen wiedergibt.

Sei A eine (m,n)-Matrix, seien f bzw. g Skalarprodukte für den $\mathbb{R}^n$ bzw. den $\mathbb{R}^m$, sei $y_o \in \mathbb{R}^m$, beliebig aber fest.

Wir betrachten die Gleichung

$$Ax = y_o \tag{13}$$

3.8 Definition

Jede Lösung $\bar{x}$ des Programms

$$\min \; \|Ax - y_o\|_g \,, \; x \in \mathbb{R}^n$$

heißt extremale Lösung von (13). Sei E (A, y_o) die Menge aller dieser extremalen Lösungen, sei x_o das eindeutig bestimmte Element minimaler Norm, d.h.

$$x_o \in E\,(A, y_o)$$

$$\forall\, x\colon x \in E(A, y_o) \Longrightarrow \|x_o\|_f \leq \|x\|_f .$$

Dann heißt die Abbildung

$$A^+ : \mathbb{R}^m \to \mathbb{R}^n$$
$$y_o \mapsto x_o \qquad \forall\, y_o \in \mathbb{R}^m$$

die Pseudoinverse von A.

Die beiden folgenden Sätze enthalten einige wichtige Eigenschaften der Pseudoinversen.

3.9 Satz

(i) $AA^+A = A$

(ii) $A^+AA^+ = A^+$

(iii) $(A^+A)^* = A^+A$

(iv) $(AA^+)^* = AA^+$

(v) $AA^+ = P_{R(A)}$

(vi) $I-A^+A = P_{N(A)}$

(vii) $A^+ = (A^*A)^+A^*$

(viii) $A^+ = A^* (AA^*)^+$.

3.10 Satz

(i) Die Gleichung (13) ist genau dann lösbar, falls gilt

$$AA^+ y_o = y_o$$

(ii) Falls (13) eine Lösung hat, so hat jede Lösung x die Form

$$x = A^+ y_o + (I-A^+A)\, z$$

für ein $z \in \mathbb{R}^n$. Umgekehrt ist jedes x dieser Form Lösung von (13).

Die Pseudoinverse liefert also ein einfaches Kriterium für die Lösbarkeit von linearen Gleichungen. Man beachte, daß im Falle der Lösbarkeit von (13) der Vektor

$$x_o = A^+ y_o$$

nach 3.8 das eindeutig bestimmte Element minimaler Norm in dem durch (13) definierten affinen Unterraum ist.

4. Die Formulierung des Problems als quadratisches Programm

Um eine einheitliche Bezeichnung zu haben, wollen wir folgende Vereinbarung treffen:

Eine Matrix X heißt die (m+n, m·n)-Transportmatrix: $\Longleftrightarrow$

X hat die Form

wobei an den nichtausgefüllten Stellen Nullen stehen.
Eine Matrix Y heißt die um die letzte Zeile verkürzte Transportmatrix: $\Longleftrightarrow$ Y entsteht aus der (m+n, m·n)-Transportmatrix durch Weglassen der letzten Zeile unter Beibehaltung aller Spalten.

Wir behandeln zunächst den Fall ohne Vorgabewerte.

Sei $A = ((a_{ij}))$ eine reelle (m,n)-Matrix, seien

$z_i > 0$, $i=1,\ldots,m$; $s_j > 0$, $j=1,\ldots,n$ reelle Zahlen mit

$$\sum_{i=1}^{m} z_i = \sum_{j=1}^{n} s_j \,.$$

Sei a_i^T die i-te Zeile von A, $a^T = (a_1^T, \ldots, a_m^T)$,
$\tilde{b}^T = (z_1, \ldots, z_m, s_1, \ldots, s_n)$, $\tilde{B}$ die (m+n, m·n)- Transportmatrix.

Im folgenden wird jeder Vektor $x \in \mathbb{R}^{m \cdot n}$ mit Doppelindizes in der Form

$$x^T = (x_{11}, \ldots, x_{1n}, x_{21}, \ldots, x_{2n}, \ldots, x_{m1}, \ldots, x_{mn})$$

geschrieben. Alle betrachteten Vektorräume, also insbesondere der $\mathbb{R}^{mn}$, werden mit dem Skalarprodukt

$$f(x,y) := x^T y$$

und der dazugehörenden Norm

$$\|x\| := \sqrt{x^T x}$$

versehen. Wir betrachten das Problem

$$\begin{aligned} \min \quad & \|x-a\| \\ \text{unter} \quad & \tilde{B}x = \tilde{b} \\ & x \geq o. \end{aligned} \tag{14}$$

Man beachte, daß für den Vektor a auch negative Komponenten zugelassen sind. Dies wird in Kapitel 6 (Vorgabewerte) wichtig sein, denn bei der Behandlung von Vorgabewerten wird a so modifiziert, daß der daraus entstehende Vektor auch bei semipositiver Ausgangsmatrix A unter Umständen negative Komponenten enthält.

Die letzte Zeile von $\tilde{B}$ ist linear abhängig von den ersten m+n-1 Zeilen. Deshalb betrachten wir im folgenden nur die um die letzte Zeile verkürzte Transportmatrix B. Sei b die entsprechende rechte Seite der Gleichung in (14).

Nach der Translation

$$\tilde{x} := x - a$$

in das globale Minimum der Zielfunktion erhält man

$$\begin{aligned} &\min && \|\tilde{x}\| \\ &\text{unter} && B\tilde{x} = b - B\cdot a \\ & && \tilde{x} \geq -a \,. \end{aligned} \tag{15}$$

Nach 3.8 ist unter allen Lösungen von $B\tilde{x}=b-Ba$ der Vektor

$$\tilde{x}_o := B^+(b-Ba)$$

das eindeutig bestimmte Element minimaler Norm. Falls gilt $\tilde{x}_o \geq -a$, so ist das Problem schon an dieser Stelle gelöst. Im allgemeinen ist dies natürlich nicht der Fall.

Jedes $\tilde{x}$ mit $B\tilde{x}=b-Ba$ kann geschrieben werden

$$\tilde{x} = \tilde{x}_o + Sy \,, \tag{16}$$

wobei S eine $(m \cdot n, (m-1) \cdot (n-1))$-Matrix ist, deren Spalten eine Basis für N(B) bilden. Es ist für unsere Zwecke nicht notwendig, eine Basis explizit anzugeben, obwohl dies mit Hilfe der Pseudoinversen möglich ist. Dies wird deshalb erst an späterer Stelle (Kapitel 5) geschehen.
Wegen (16) ist (15) äquivalent zu

$$\min \ \|\tilde{x}_o\|^2 + 2 \cdot \tilde{x}_o^T Sy + y^T S^T Sy \tag{17}$$

$$\text{unter } S_y \geq - \tilde{x}_o - a \ .$$

Sei $x_o := \tilde{x}_o + a$. Nach Konstruktion ist $\tilde{x}_o \perp N(B)$, deshalb erhält man aus (17)

$$\min \ y^T S^T Sy \tag{18}$$

$$\text{unter } Sy \geq - x_o \ .$$

Wegen $\Sigma z_i = \Sigma s_j$ existiert ein $\bar{x} > o$ mit $B\bar{x} = b$, also existiert auch ein $\bar{y} \in \mathbb{R}^{(m-1)(n-1)}$, so daß $S\bar{y} > -x_o$. Damit ist die Slater-Bedingung erfüllt, und man kann das zu (18) duale Programm betrachten. Da S maximalen Spaltenrang hat, ist $S^T S$ positiv definit, also insbesondere invertierbar. Deshalb lautet das duale Programm

$$\min \ \tfrac{1}{4} \lambda^T S \, (S^T S)^{-1} S^T \lambda + x_o^T \lambda \tag{19}$$

$$\text{unter } \lambda \geq o.$$

Als nächstes wollen wir die Matrix $S(S^TS)^{-1}S^T$ umformen und benutzen dazu die Pseudoinverse S^+ von S. Wegen 3.7 und 3.9 (vii) hat S^+ die Darstellung

$$S^+ = (S^TS)^{-1} S^T.$$

Nach Konstruktion ist R(S) = N(B). Wegen 3.9 (V) und 3.9 (vi) gilt

$$SS^+ = P_{R(S)} \quad \text{und} \quad I-B^+B = P_{N(B)},$$

also erhält man

$$\begin{aligned} S(S^TS)^{-1}S^T &= SS^+ \\ &= P_{R(S)} \\ &= P_{N(B)} \\ &= I-B^+B. \end{aligned}$$

Mit der Abkürzung $P := I - B^+B$ lautet (19) also

$$\min_{\lambda} \frac{1}{4} \lambda^T P\lambda + x_o^T\lambda \qquad (20)$$

$$\text{unter } \lambda \geq o.$$

Sei $\bar{\lambda}$ ein Optimalpunkt von (20). Daraus erhält man den eindeutig bestimmten optimalen Punkt $\bar{x}$ des primalen Programms (14) als

$$\bar{x} = x_o + \frac{1}{2} P \bar{\lambda}.$$

Um Problem (20) und damit (14) zu lösen, müssen also noch B^+ und P berechnet werden. Dies wird im folgenden Kapitel geschehen.

5. Die Pseudoinverse der Transportmatrix

Es gibt eine große Anzahl von Verfahren zur Bestimmung der Speudoinversen einer Matrix. Einige davon findet man zum Beispiel in [2] . Die numerische Pseudoinversion ist jedoch aus drei Gründen nicht empfehlenswert. Zunächst benötigen B^+ und P sehr viel Platz in einer Rechenanlage (B^+ hat die Größe (m·n, m+n-1), P die Größe (m·n, m·n); dies ergibt bei einem in der Praxis durchaus vorkommenden Wert von m=n=50 247500 Speicherplätze für B^+ und 6250000 für P). Zum anderen ist bekannt, daß die Abbildung, die jeder Matrix ihre Pseudoinverse zuordnet, nicht stetig ist. Dies wirkt sich zum Beispiel dann aus, wenn zwei Matrizen sich zwar elementweise wenig unterscheiden, jedoch verschiedene Ränge haben.

Ein einfaches Beispiel ist

$$D_1 = \begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}, \quad D_2 = \begin{bmatrix} 2 & 0 \\ 0 & 10^{-10} \end{bmatrix}$$

mit

$$D_1^+ = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix}, \quad D_2^+ = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 10^{10} \end{bmatrix} .$$

Die gängigen Verfahren zur Pseudoinversion bestimmen implizit den Rang der Matrix. Deshalb kann es passieren, daß auf Grund von Rundungsfehlern das Ergebnis des Algorithmus nicht die "richtige" Pseudoinverse ist. Drittens müssen im dualen Programm (20) bei jedem Iterationsschritt Vektoren der Form $P\lambda$ berechnet werden. Wenn man B nur numerisch pseudoinvertiert, hat man keine Kenntnis über die Struktur von P, so daß die gesamte Iteration für große m und n unter

Umständen sehr lange dauert. Wie sich später erweisen wird (Korollar 5.5, 5.6), hat P jedoch eine einfache Blockstruktur, die eine leichte Berechnung von $P\lambda$ gestattet. Deshalb wollen wir im folgenden eine explizite Formel für B^+ herleiten. Dazu sei zunächst ein aus der linearen Algebra bekanntes Lemma vorausgeschickt.

5.1 Lemma

Sei A eine positiv definite Matrix, die partitioniert ist in der Form

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{matrix} \} P \\ \} q \end{matrix}$$

$$\underbrace{\phantom{A_{11}}}_{P} \quad \underbrace{\phantom{A_{12}}}_{q}$$

Sei $Q := A_{22} - A_{21} A_{11}^{-1} . A_{12}$ regulär. Dann hat A^{-1} die Form

$$A^{-1} = \begin{bmatrix} A_{11}^{-1} + A_{11}^{-1} A_{12} Q^{-1} A_{21} A_{11}^{-1} & -A_{11}^{-1} A_{12} Q^{-1} \\ - Q^{-1} A_{21} A_{11}^{-1} & Q^{-1} . \end{bmatrix} \quad (21)$$

Sei I_1 die (1,1)-Einheitsmatrix, E_{1k} diejenige (1,k)-Matrix, deren Elemente alle gleich Eins sind, $e^T = (1,\ldots,1)$, wobei die Anzahl der Komponenten von e aus dem Zusammenhang hervorgeht.

5.2 Satz

Sei $\tilde{B}$ die (m+n, mn)-Transportmatrix, B die um die letzte Zeile verkürzte Transportmatrix. Die (n,n-1)-Matrix Y sei definiert durch

$$Y = \frac{1}{m} \cdot \begin{bmatrix} E_{n-1,\,n-1} \\ -e^T \end{bmatrix}$$

Für jedes i=1,...,m sei X_i die (n,m)-Matrix

$$X_i := \frac{1}{mn} \cdot \left[\begin{array}{c:c:c} E_{n-1,i-1} & (m-1)e & E_{n-1,m-i} \\ \hdashline (n-1)e^T & m+n-1 & (n-1)e^T \end{array}\right]$$

$$\uparrow$$

Spalte i

Dann hat B^+ die Form

$$B^+ = \begin{bmatrix} X_1 & Y \\ X_2 & Y \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ X_m & Y \end{bmatrix}$$

Beweis:

Nach 3.7 und 3.9 (viii) hat B^+ die Darstellung

$$B^+ = B^T (BB^T)^+ .$$

Da B vollen Zeilenrang hat, ist BB^T positiv definit, und deshalb ist

$$(BB^T)^+ = (BB^T)^{-1} .$$

BB^T hat die Form

$$BB^T = \begin{bmatrix} nI_m & E_{m,n-1} \\ E_{n-1,m} & mI_{n-1} \end{bmatrix} \qquad (22)$$

Hier ist mit den Bezeichnungen von 5.1

$$Q = mI_{n-1} - \frac{m}{n} \cdot E_{n-1,n-1}$$

Wie man leicht sieht, ist Q invertierbar, und zwar gilt

$$Q^{-1} = \frac{1}{m} \cdot (E_{n-1,n-1} + I_{n-1}) .$$

Einsetzen der entsprechenden Ausdrücke aus (22) in (21) und Vormultiplikation mit B^T liefert dann die Behauptung. ⊣

Mit Hilfe von 5.2 und der folgenden von Greville [11] bewiesenen Aussage kann man nun die Pseudoinverse der vollen (m+n, m.n)-Transportmatrix bestimmen.

5.3 Lemma (Greville [11]).

Für jede (m,n)-Matrix A und jedes $a \in \mathbb{R}^n$ ist

$$\begin{bmatrix} A \\ a^T \end{bmatrix}^+ = ((I-ba^T)\cdot A^+, b),$$

wobei

$$b := \begin{cases} \dfrac{A^+A^{+T}a}{1+\| A^{+T}a\|^2} & \text{, falls a Linearkombination der Zeilen von A} \\ \dfrac{(I-A^+A)\,a}{\|(I-A^+)a\|^2} & \text{sonst} \end{cases}$$

Beachtet man, daß die letzte Zeile von $\tilde{B}$ von den anderen Zeilen linear abhängig ist, so erhält man also:

5.4 Korollar

Sei $\tilde{B}$ die (m+n, m.n)-Transportmatrix, sei $\tilde{Y}$ die (n,n)-Matrix

$$\tilde{Y} := \frac{1}{m(m+n)} \cdot ((m+n)I_n - E_{n,n}).$$

Für jedes i=1,...,m sei $\tilde{X}_i$ die (n,m)-Matrix

$$X_i := \frac{1}{n(m+n)} \cdot (-E_{n,i-1} \vdots (m+n-1)e \vdots - E_{n,m-i})$$

$\uparrow$
Spalte i

Dann wird die Pseudoinverse von $\tilde{B}$ gegeben durch

$$\tilde{B}^+ = \begin{bmatrix} \tilde{X}_1 & \tilde{Y} \\ \tilde{X}_2 & \tilde{Y} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \tilde{X}_m & \tilde{Y} \end{bmatrix}$$

5.5 Korollar

Sei B die um die letzte Zeile verkürzte Transportmatrix, sei $P := I - B^+B$. Die (n,n)-Matrizen U und V seien definiert durch

$$U := \frac{m-1}{mn} \cdot (nI_n - E_{n,n})$$

$$V := -\frac{1}{m-1} \cdot U$$

Dann gilt

$$P = \begin{bmatrix} U & V & \cdots & \cdots & V \\ V & U & & & \\ & & & & \\ & & & & V \\ V & & & V & U \end{bmatrix}$$

m-1 mal

Der Beweis folgt sofort aus 5.2.

Bemerkung: Wegen $N(B) = N(\tilde{B})$ gilt natürlich

$$I - B^+B = I - \tilde{B}^+\tilde{B}.$$

Die Aussagen 5.2 und 5.5 gestatten es nun, die Komponenten für die im dualen Programm (20) auftretenden Vektoren $P\lambda$ und für den Vektor $x_o = B^+b + Pa$ explizit anzugeben, ohne B^+ und P speichern zu müssen.

4.6 Korollar

Für jedes $\lambda \in \mathbb{R}^{mn}$ hat der Vektor $P\lambda$ die Komponenten

$$(P\lambda)_{ij} = \lambda_{ij} - \frac{1}{m} \cdot \sum_{k=1}^{m} \lambda_{kj} - \frac{1}{n} \cdot \sum_{l=1}^{n} \lambda_{il} + \frac{1}{mn} \cdot \sum_{k=1}^{m} \sum_{l=1}^{n} \lambda_{kl}$$

Beweis: In der Darstellung

$$\lambda^T = (\lambda_{11}, \ldots, \lambda_{1n}, \ldots, \lambda_{m1}, \ldots, \lambda_{mn})$$

für jedes $\lambda \in \mathbb{R}^{mn}$ setzen wir zur Abkürzung

$$\lambda_i^T = (\lambda_{i1}, \ldots, \lambda_{in})$$

für jedes $i = 1, \ldots, m$.

Dann ist

$$(P\lambda)_i = U\,\lambda_i + V \left(\sum_{\substack{k=1 \\ k \neq i}}^{m} \lambda_k \right)$$

$$= \frac{m-1}{mn}(nI-E)\lambda_i - \frac{1}{mn}(nI-E)\Big(\sum_{\substack{k=1\\k\neq i}}^{m}\lambda_k\Big),$$

wobei $I = I_n$, $E = E_{n,n}$

$$= \frac{1}{mn}(nI-E)\Big((m-1)\lambda_i - \sum_{\substack{k=1\\k\neq i}}^{m}\lambda_k\Big)$$

$$= \frac{1}{mn}\Big(nmI\lambda_i - n\cdot\sum_{k=1}^{m}\lambda_k - mE\lambda_i + E\cdot\sum_{k=1}^{m}\lambda_k\Big).$$

Sei $u_i := \sum_{l=1}^{n}\lambda_{il}$, $v_j := \sum_{k=1}^{m}\lambda_{kj}$, $\sigma := \sum_{i=1}^{m}u_i = \sum_{j=1}^{n}v_j$.

Dann ist

$$\Big(\sum_{k=1}^{m}\lambda_k\Big)^T = (v_1, v_2, \ldots, v_n)$$

$$(E\lambda_i)^T = (u_i, \ldots, u_i)$$

$$\Big(E\cdot\sum_{k=1}^{m}\lambda_k\Big)^T = (\sigma, \ldots, \sigma).$$

Daraus folgt die Behauptung. ⊣

5.7 Korollar

Der Vektor $x_o = B^+b + Pa$ hat die Komponenten

$$(x_o)_{ij} = a_{ij} + \frac{1}{m}\Big(s_j - \sum_{k=1}^{m}a_{kj}\Big) + \frac{1}{n}\Big(z_i - \sum_{l=1}^{n}a_{il}\Big) + \frac{1}{mn}\Big(e^TAe - \sum_{i=1}^{m}z_i\Big)$$

Beweis: Mit $z^T=(z_1,\ldots,z_m)$, $s^T=(s_1,\ldots,s_n)$ gilt

$$(B^+b)_i = X_i z + Ys \ .$$

Sei $\sigma_1 := \sum_{k=1}^{m} z_k$, $\sigma_2 := \sum_{l=1}^{n-1} s_l$. Dann ist

$$(X_i z)^T = \Big(\underbrace{\frac{z_i}{n} - \frac{\sigma_1}{mn}, \ldots, \frac{z_i}{n} - \frac{\sigma_1}{mn}}_{\text{n-1 mal}} + \frac{\sigma_1}{m} \Big)$$

und

$$(Ys)^T = \frac{1}{m} \cdot (s_1, s_2, \ldots, s_{n-1}, -\sigma_2),$$

also

$$(B^+b)_{ij} = \begin{cases} \dfrac{z_i}{n} - \dfrac{\sigma_1}{mn} + \dfrac{s_j}{m} & \text{für } j \neq n \\[2ex] \dfrac{z_i}{n} - \dfrac{\sigma_1}{mn} + \dfrac{1}{m}(\sigma_1 - \sigma_2) & \text{für } j = n \end{cases}$$

Wegen $\sum_{k=1}^{m} z_k = \sum_{l=1}^{n} s_l$ ist $\sigma_1 - \sigma_2 = s_n$. Berücksichtigt man noch die Darstellung aus 5.6 für die Komponenten des Vektors Pa, so folgt die Behauptung. ⊣

Zum Schluß wollen wir mit Hilfe der Pseudoinversen explizit eine Basis für den Kern der (m+n, m·n)-Transportmatrix angeben. Dazu benötigen wir folgendes Lemma über die Determinante von Blockmatrizen einer bestimmten Gestalt.

5.8 Lemma

Seien X,Y (n,n)-Matrizen, sei A die (m·n, m·n)-Matrix

$$A = \underbrace{\begin{bmatrix} X & Y & \cdots & Y \\ Y & X & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & Y \\ Y & \cdots & Y & X \end{bmatrix}}_{m-1 \text{ mal}}$$

Dann gilt für die Determinante $|A|$ von A:

$$|A| = |X - Y|^{m-1} \, |X + (m-1)\, Y|.$$

Beweis:

Durch elementare Spaltenoperationen erhält man

$$|A| = \begin{vmatrix} X+(m-1)Y & (m-1)Y & (m-2)Y & \cdots & 2Y & Y \\ \cdot & X+(m-2)Y & (m-2)Y & \cdots & 2Y & \cdot \\ \cdot & X+(m-2)Y & x+(m-3)Y & \cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & & \cdot & \cdot \\ \cdot & \cdot & \cdot & & X+Y & Y \\ \overline{X+(m-1)Y} & \overline{X+(m-2)Y} & \overline{X+(m-3)Y} & & \overline{X+Y} & \overline{X} \end{vmatrix}$$

Bezeichnen wir in dieser Darstellung die aus den ersten $(m-1)\cdot n$ Zeilen und Spalten bestehende Teilmatrix mit B, die aus den ersten $(m-1)\cdot n$ Zeilen und n letzten Spalten bestehende Teilmatrix mit C, so erhält man durch elementare Zeilenoperationen (Subtraktion der Zeilen $(m-2)n+j$ von den Zeilen $(m-1)n+j$ für $j=1,\ldots,n$):

$$|A| = \begin{vmatrix} B & C \\ 0 & X-Y \end{vmatrix} = |B|\,|X-Y|.$$

Durch wiederholte Anwendung des Verfahrens folgt dann die Behauptung. ⊣

Nun gilt nach 3.10

$$\tilde{B}\,x = o \Longleftrightarrow x = Pz \quad \text{für ein } z \in \mathbb{R}^{mn}.$$

Also sind die linear unabhängigen Spalten von P eine Basis für $N(B) = N(\tilde{B})$. Nach 5.5 hat P die Darstellung

$$P = (P_1, P_2, \ldots, P_m),$$

mit

$$P_i = \begin{bmatrix} V \\ V \\ \cdot \\ \cdot \\ \cdot \\ U \\ \cdot \\ \cdot \\ V \end{bmatrix} \leftarrow \text{i-ter Block, } i=1,\ldots,m \qquad (23)$$

5.9 Satz

Für i=1,...,m-1 sei S_i diejenige (m n, n-1)-Matrix, die in der Darstellung (23) aus P_i durch Streichen der letzten Spalte entsteht. Dann bilden die Spalten der Matrix

$$S: = (S_1, S_2, \ldots, S_{m-1})$$

eine Basis für den Kern von $\tilde{B}$.

Beweis:

Seien U_o, V_o die durch

$$U_o: = (m-1)(nI_{n-1} - E_{n-1, n-1})$$

$$V_o: = -\frac{1}{m-1} \cdot U_o$$

definierten (n-1, n-1)-Matrizen.

Dann genügt es zu zeigen, daß die Matrix

$$S_o = \underbrace{\begin{bmatrix} U_o & V_o & \cdots & V_o \\ V_o & U_o & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \vdots \\ & & & V_o \\ V_o & & V_o & U_o \end{bmatrix}}_{m-2 \text{ mal}}$$

regulär ist.

Nach 5.8 ist

$$|S_o| = |U_o - V_o|^{m-2} |U_o + (m-2)\, V_o|.$$

Nun ist

$$U_o - V_o = \frac{m}{m-1} \cdot U_o \,,$$

$$U_o + (m-2)V_o = \frac{1}{m-1} \cdot U_o,$$

also gilt

$$|S_o| \neq o \Longleftrightarrow |U_o| \neq o.$$

U_o hat ebenfalls die Form der Matrix in 5.8, also ist

$$|U_o| = (m-1)^{n-1}\, n^{n-2}.$$

Daraus folgt die Behauptung.⊣

6. Berücksichtigung von Vorgabewerten

Häufig wird verlangt, daß in der Lösung des Nichtnegativen Matrixproblems einige Komponenten fest vorgegebene Werte annehmen. Während bei Problemen ohne Vorgabewerte (Kapitel 4) wegen

$$\sum_{i=1}^{m} z_i = \sum_{j=1}^{m} s_j$$

stets zulässige Lösungen existieren, braucht dies hier natürlich nicht mehr der Fall zu sein. Wenn jedoch überhaupt ein zulässiger Punkt existiert, so auch eine zulässige Ecke des durch die Bedingungen (i) bis (iv) aus Kapitel 1 beschriebenen Polyeders, und dies läßt sich bekanntlich leicht nachprüfen, siehe zum Beispiel Vogel [30], Kapitel X. Deshalb gehen wir im folgenden davon aus, daß der zulässige Bereich nicht leer ist. Zur Lösung des Problems mit Vorgabewerten unter Benutzung der Pseudoinversen gibt es prinzipiell mehrere Möglichkeiten, die nun besprochen werden sollen. Dazu sei $M = \{1,\ldots, m\}$, $N = \{1,\ldots,n\}$, $R \subset M \times N$ die Menge aller Indexpaare, für die Vorgabewerte existieren, für $(i,j) \in R$ sei $c_{ij} \geq 0$ der entsprechende Vorgabewert.

Ein naheliegender Ansatz besteht darin, die Nebenbedingungen

$$x_{ij} = c_{ij} \quad \forall (i,j) \in R$$

zusammenzufassen zu einer Gleichung $Cx=c$ mit einer Null-Eins Matrix C der Größe $(|R|, m \cdot n)$ und einem Vektor $c \in \mathbb{R}^{|R|}$. Damit erhält man

$$\begin{aligned} &\min \ \|x-a\| \\ &\text{unter } Bx=b \\ &\qquad Cx=c \\ &\qquad x \geq 0 \end{aligned} \tag{24}$$

Sei

$$D := \begin{bmatrix} B \\ C \end{bmatrix}, \quad d := \begin{bmatrix} b \\ c \end{bmatrix}.$$

In Analogie zu dem Vorgehen in Kapitel 4 betrachtet man die Pseudoinverse von D. Dann hat jede Lösung von (24) die Form

$$x = D^+d + (I-D^+D)y$$

für ein passendes $y \in \mathbb{R}^{mn}$. Nach Cline [7] hat D^+ die Form

$$D^+ = (B^+-G\cdot C\cdot B^+, G),$$

wobei

$$G=F^+ +(I-F^+C)B^{+T}B^{+T}C^TH(I-FF^+)$$

$$F=C(I-B^+B)$$

$$H=\{I+[(I-FF^+)CB^+]\cdot[(I-FF^+)CB^+]^T\}^{-1}.$$

Wegen der komplizierten Form von D^+ scheidet dieser Ansatz für die Praxis aus.

Als weitere Möglichkeit betrachtet man dieselben Nebenbedingungen wie in (24). Da die Pseudoinverse von B schon bekannt ist, läßt sich die Gleichung Bx=b mit Hilfe der Darstellung $x=B^+b+Py$ aus (24) eliminieren. Damit erhält man als neue Nebenbedingungen

$$\begin{aligned} CPy &= c - CB^+b \\ Py &\geq -B^+b. \end{aligned}$$

Dies führt auf das Problem, die Pseudoinverse von CP zu bestimmen. Nach Cline [8] gilt

$$(CP)^+ = P_o^+ \cdot C_o^+ ,$$

wobei

$$P_o = C^+CP$$
$$C_o = C\, P_o P_o^+ .$$

Daraus ist ersichtlich, daß auch dieser Ansatz für praktische Belange keinen Erfolg verspricht. Der im folgenden beschriebene Ansatz bietet jedoch eine Möglichkeit, diese Schwierigkeiten zu vermeiden.

Für jedes $i \in M$ sei

$$R_i := \{j \in N \mid (i,j) \in R\},$$

für jedes $j \in N$ sei

$$R_j^{-1} := \{i \in M \mid (i,j) \in R\},$$

sei

$$a'_{ij} := \begin{cases} a_{ij} - c_{ij} & \text{falls } (i,j) \in R \\ a_{ij} & \text{sonst} \end{cases}$$

$$z'_i := z_i - \sum_{j \in R_i} c_{ij} \qquad \forall\, i \in M \qquad (25)$$

$$s'_j := s_j - \sum_{i \in R_j^{-1}} c_{ij} \qquad \forall\, j \in N .$$

Wir betrachten die beiden Probleme

$$(*)\quad \begin{cases} \min g(x) = \sum\limits_{i,j} (x_{ij}-a_{ij})^2 \\ \text{unter } \sum\limits_{j=1}^{n} x_{ij} = z_i \qquad \forall\, i \\ \sum\limits_{i=1}^{m} x_{ij} = s_j \qquad \forall\, j \\ x_{ij} = c_{ij} \qquad (i,j)\in R \\ x_{ij} \geq 0 \end{cases}$$

und

$$(**)\quad \begin{cases} \min h(x) = \sum\limits_{i,j} (x_{ij}-a'_{ij})^2 \\ \text{unter } \sum\limits_{j=1}^{n} x_{ij} = z'_i \qquad \forall\, i \\ \sum\limits_{i=1}^{m} x_{ij} = s'_j \qquad \forall\, j \\ \sum\limits_{(i,j)\in R} x_{ij} = 0 \\ x_{ij} \geq 0 \end{cases}$$

6.1 Lemma

Sei x zulässig für (*). Dann ist der Vektor y mit

$$y_{ij} := \begin{cases} 0 & \text{falls } (i,j)\in R \\ x_{ij} & \text{sonst} \end{cases}$$

zulässig für (**). Ist umgekehrt y zulässig für (**), so ist x mit

$$x_{ij} := \begin{cases} c_{ij} & \text{falls } (i,j) \in R \\ y_{ij} & \text{sonst} \end{cases}$$

zulässig für (*).

Beweis:

Sei x zulässig für (*). Dann ist

$$\sum_{j \in N} y_{ij} = \sum_{j \in R_i} y_{ij} + \sum_{j \notin R_i} y_{ij}$$

$$= o + \sum_{j \notin R_i} x_{ij}$$

$$= z_i' .$$

Sei y zulässig für (**). Dann ist

$$\sum_{j \in N} x_{ij} = \sum_{j \in R_i} x_{ij} + \sum_{j \notin R_i} x_{ij}$$

$$= \sum_{j \in R_i} c_{ij} + \sum_{j \notin R_i} y_{ij}$$

Wegen $\sum_{(i,j) \in R} y_{ij} = o$ und $y_{ij} \geq o \ \forall\, i,j$ ist $y_{ij} = o \ \forall\, (i,j) \in R$

Also ist $\sum_{j \in N} y_{ij} = \sum_{j \notin R_i} y_{ij}$, und daher

$$\sum_{j \in N} x_{ij} = \sum_{j \in R_i} c_{ij} + z_i - \sum_{j \in R_i} c_{ij} = z_i .$$

Analog zeigt man, daß die Forderung über die Spaltensummen erfüllt ist. ⊣

6.2 Lemma

Die Punkte x und y seien wie in der Voraussetzung von 6.1. Dann gilt

$$x \text{ optimal für } (*) \iff y \text{ optimal für } (**).$$

Beweis: Falls y nicht optimal ist, so existiert ein zulässiges y' mit h(y') < h(y). Nach 6.1 ist dann x' mit

$$x'_{ij} := \begin{cases} c_{ij} & \text{für } (i,j)\in R \\ y'_{ij} & \text{sonst} \end{cases}$$

zulässig für (*). Aus h(y') < h(y) erhält man g(x')<g(x), also ist x nicht optimal für (*). Die Umkehrung folgt analog. ⊣

Von nun an setzen wir voraus, daß die a_{ij}, z_i und s_j gemäß (25) modifiziert sind und lassen deshalb die Striche fort. Wegen 6.1 und 6.2 betrachten wir das Problem

$$\begin{aligned} &\min \ \|x-a\| \\ &\text{unter } Bx=b \\ &\qquad r^T x=o \\ &\qquad x \geq o \ , \end{aligned} \tag{26}$$

wobei der Vektor $r\in\mathbb{R}^{mn}$ definiert ist durch

$$r_{ij} := \begin{cases} 1 & \text{falls } (i,j)\in R \\ o & \text{sonst.} \end{cases}$$

Man beachte, daß a durch die obige Modifikation negative Komponenten haben kann. Für (26) ist natürlich nur der Fall interessant, daß r linear unabhängig von den Zeilen von B ist. Dies ist leicht nachzuprüfen, denn r ist genau dann Linearkombination der Zeilen von B, wenn $r \in R(B^T)$, und

$$\begin{aligned} r \in R(B^T) &\Longleftrightarrow r \in [N(B)]^{\perp} \\ &\Longleftrightarrow Pr = o \\ &\Longleftrightarrow mnr_{ij} - n \cdot \sum_{k=1}^{m} r_{kj} - m \cdot \sum_{l=1}^{n} r_{il} + e^T r = o \end{aligned}$$

für alle i,j (siehe 5.6). Wir nehmen im folgenden $Pr \neq o$ an.

Sei

$$D := \begin{bmatrix} B \\ r^T \end{bmatrix}, \quad d := \begin{bmatrix} b \\ o \end{bmatrix},$$

$$y_o := D^+ d + (I - D^+ D)a, \quad p := (m-1)(n-1) - 1.$$

Dann erhält man analog zu (15) das Problem

$$\begin{aligned} &\min \ \|\tilde{x}\| \\ &\text{unter } D\tilde{x} = d - Da, \ \tilde{x} \geq o \end{aligned} \tag{27}$$

und daraus analog zu (18)

$$\begin{aligned} &\min \ v^T S^T S \, v \\ &\text{unter } Sv \geq -y_o \ , \end{aligned} \tag{28}$$

wobei $v \in \mathbb{R}^p$ und S eine $(m \cdot n, p)$-Matrix ist, deren Spalten eine Basis für N(D) bilden. Durch die Dimensionsreduzierung von mn auf p ist die Slater-Bedingung für (28) erfüllt, denn

$$\dim \{\tilde{x} \mid D\tilde{x} = d - Da,\ \tilde{x} \geq -a\} = p,$$

also hat der zulässige Bereich von (28) innere Punkte.
Mit der Abkürzung

$$Q := I - D^{+}D$$

lautet das duale Programm

$$\min \ \frac{1}{4}\lambda^{T} Q\lambda + y_{o}^{T}\lambda \qquad (30)$$

$$\text{unter } \lambda \geq o\ .$$

Wenn $\bar{\lambda}$ optimal für (30) ist, so ist $\bar{x} := y_{o} + \frac{1}{2}Q\bar{\lambda}$ optimal für (26). Es bleibt noch die Berechnung von y_{o} und Q. Wegen 5.3 erhält man mit $P = I - B^{+}B$ und $x_{o} = B^{+}b + Pa$

$$y_{o} = x_{o} - \|Pr\|^{-2}\, Prr^{T}x_{o}$$

$$Q = P - \|Pr\|^{-2}\, (Pr)(Pr)^{T}.$$

Es ist nicht notwendig, in einer Rechenanlage die Matrix $(Pr)(Pr)^{T}$ zu speichern, sondern nur den Vektor Pr, da in (30) bei der zyklischen Minimierung nur der Ausdruck

$$Q\lambda = P\lambda - \|Pr\|^{-2}\ Pr\ \alpha(\lambda)$$

mit dem von λ abhängigen Skalar

$$\alpha(\lambda) := r^{T}P\lambda$$

benötigt wird.

7. Numerische Ergebnisse

Die dualen Programme (20) bzw. (30) wurden jeweils mit der Methode des zyklischen Koordinatenabstiegs gelöst. Die Ausgangspunkte für die Iteration wurden wie folgt gewählt:

Sei e^{ij} der Vektor des $\mathbb{R}^{mn}$ mit Koordinaten

$$(e^{ij})_{kl} = \begin{cases} 1 & \text{falls } k=i \text{ und } l=j \\ 0 & \text{sonst} \end{cases},$$

sei λ_o der Startpunkt der Iteration. Im Falle des Programms (20) wurde λ_o definiert durch

$$(\lambda_o)_{ij} := \begin{cases} -2\,\dfrac{(x_o)_{ij}}{e^{ij^T} P e^{ij}} & \text{falls } (x_o)_{ij} < o \\ 0 & \text{sonst} \end{cases},$$

und im Falle des Programms (30) analog durch

$$(\lambda_o)_{ij} := \begin{cases} -2\,\dfrac{(y_o)_{ij}}{e^{ij^T} Q e^{ij}} & \text{falls } (y_o)_{ij} < o \\ 0 & \text{sonst.} \end{cases}$$

Alle Tests wurden auf einer IBM 370/168 durchgeführt. Als Vergleich für die erhaltenen Rechenzeiten diente der Algorithmus von Beale in der in [15] veröffentlichten Form.

Zum Testen wurden einmal zufällig erzeugte und zum anderen in der Literatur vorliegende Daten verwendet.

7.1 Zufällig erzeugte Probleme

Zur Erstellung der neuen Zeilen- und Spaltensummen wurden zunächst im Intervall (0,1) gleichverteilte Zahlen z_i', $i=1,\ldots,m$; s_j', $j=1,\ldots,n$ für verschiedene m und n generiert.

Mit den Abkürzungen

$$\alpha := \frac{\sum_{i=1}^{m} z_i' + \sum_{j=1}^{n} s_j'}{\sum_{i=1}^{m} z_i'}$$

$$\beta := \frac{\sum_{i=1}^{m} z_i' + \sum_{j=1}^{n} s_j'}{\sum_{j=1}^{n} s_j'}$$

wurden die neuen Zeilensummen z_i und die neuen Spaltensummen s_j definiert durch

$$z_i := \alpha\, z_i' \qquad \forall\, i$$

$$s_j := \beta\, s_j' \qquad \forall\, j$$

Sei $b^T = (z_1,\ldots, z_m, s_1,\ldots, s_n)$. Für festes b wurden anschließend die Elemente a_{ij} der Ausgangsmatrix A so bestimmt, daß die dazugehörigen m·n-Vektoren a auf der Sphäre mit Radius

$$r = \rho \cdot \|B^+ b\|$$

gleichverteilt waren. Der Parameter ρ wurde dabei im Intervall $[1,20]$ beliebig gewählt. (Man beachte, daß $\| B^{+}b \|$ der Abstand vom Nullpunkt zu dem durch b und die Transportmatrix B bestimmten affinen Unterraum ist.) Da für zufällig erzeugte rechte Seiten b die Forderung, daß in der Lösung bestimmte Vorgabewerte angenommen werden müssen, i.a. die Zulässigkeit zerstört, wurde für die so gewählten Beispiele nur die Problemklasse aus Kapitel 4 behandelt.

Die größten betrachteten Matrizen hatten 50 Zeilen und 50 Spalten. Die Iteration wurde abgebrochen, falls zwei aufeinanderfolgende Iterationsvektoren sich komponentenweise um höchstens 10^{-5} unterschieden. Dabei ergaben sich folgende durchschnittliche Rechenzeiten:

m=n=50, ρ = 1 :	6.19	Sek.
m=n=50, ρ = 10 :	12.30	Sek.
m=n=50, ρ = 20 :	15.09	Sek.

Für den Beale'schen Algorithmus [15] wurde als obere Schranke für die Rechenzeit 20 Minuten gesetzt. Es erwies sich, daß das Verfahren bei jedem der oben genannten m, n, ρ nach 20 Minuten reiner CPU-Zeit abbrach, ohne den optimalen Punkt gefunden zu haben. Ein weiteres Problem bot der benötigte Speicherplatz. Während bei Benutzung der Pseudoinversen für die dimensionierten Variablen insgesamt

$$5\ mn\ +\ 2\ (m+n).$$

Speicherplätze benötigt werden (siehe Programmbeschreibungen in folgendem Anhang), ist der benötigte Speicherplatz bei anderen Verfahren der quadratischen Optimierung i.a. wesentlich höher. Der zum Beispiel beim Beale'schen Verfahren in der Fassung [15] notwendige Kernspeicherplatz ergibt sich wie folgt:

Sei

N: = Maximalzahl aller auftretenden Nebenbedingungen
I: = Maximale Zeilenzahl aller auftretenden Basismatrizen
L: = Maximalzahl aller von Null verschiedenen Elemente in der Nebenbedingungsmatrix

Dann werden

$$9\ mn + 8\ N + 5\ I + I^2 + 2\ L + 9$$

Speicherplätze für die dimensionierten Variablen benötigt.

Für ein aktuelles Problem ist i.a. vor Beginn der Rechnung nicht vorhersagbar, wie groß N, I, L werden.

Trivialerweise gilt jedoch für die hier vorliegende Problemklasse

$$N \geq m+n-1$$
$$I \geq m+n-1$$
$$L \geq 2mn-m$$

Daraus ergibt sich als minimaler Speicherbedarf

$$15\ mn + 9m + 11n + m^2 + n^2 - 3\ .$$

7.2 Probleme mit in der Literatur veröffentlichten Daten

Quellen:

R. Krengel, J. Schrintke, R. Stäglin: Jährliche nominale Input-Output- Tabellen und Importmatrizen für die BRD 1954 - 1967
DIW-Beiträge zur Strukturforschung, Heft 21, Berlin 1972

W. Kirner: Investitionsverflechtung in der BRD 1960-1967
DIW, Berlin 1967

Hierbei handelt es sich um Matrizen der Größe 38x45. Als Ausgangsmatrix wurden die Matrix des Jahres 1954 gewählt, als neue Zeilen- und Spaltensummen sukzessiv die Zahlen der Jahre 1955 bis 1967. Ohne Vorgabewerte ergaben sich als untere und obere Schranken für die Rechenzeiten:

von 1954 auf 1955:	11.08	Sek.
von 1954 auf 1967:	14.30	Sek.

Bei Einführung von Vorgabewerten wurden die Rechenzeiten nicht unbeträchtlich erhöht. Es ergab sich:

von 1954 auf 1955:	118.08	Sek.
von 1954 auf 1967:	214.69	Sek.

Es stellte sich jedoch heraus, daß in diesen Fällen der Algorithmus von Beale sogar nach 30 Munuten reiner CPU-Zeit noch nicht fertig war, so daß auch hier eine Verbesserung festgestellt werden kann.

A N H A N G

Fortran-Programme

Die Programme zur Lösung des nichtnegativen Matrixproblems sind ein- und ausgabeunabhängig als Subroutines geschrieben. Alle verwendeten Größen sind dabei doppelt genau. MESTP löst das Problem ohne, MESTPV das Problem mit Vorgabewerten.

Subroutine MESTP

Der Name ist eine Abkürzung für "Minimierung des euklidischen Skalarprodukts über einem Transportpolyeder".

Zweck

Lösung des Nichtnegativen Matrixproblems ohne Vorgabewerte als Normminimierungsaufgabe bezüglich der euklidischen Norm.

Methode

Lösung des dualen Programms durch zyklischen Koordinatenabstieg.

Aufruf

CALL MESTP(A, XO, X, XA, OPT, Z, S, U, V, TOL, OBJ, IT, M,N

Beschreibung der Parameter

A Vektor der Länge M*N. Enthält zeilenweise die Elemente der Ausgangsmatrix A.

XO Vektor der Länge M*N. Enthält das Minimum der Zielfunktion des primalen Programms ohne Vorzeichenrestriktionen.

X Vektor der Länge M*N. Laufender Iterationsvektor.

XA Vektor der Länge M*N. Während des Programmablaufs Hilfsspeicher. Nach Ende der Rechnung dienen die Elemente von XA zur Überprüfung auf die Kuhn-Tucker-Bedingungen (siehe Bemerkungen unten).

OPT Vektor der Länge M*N. Optimaler Punkt des primalen Problems.

Z Vektor der Länge M. Enthält die vorgegebenen (neuen) Zeilensummen.

S Vektor der Länge N. Enthält die vorgegebenen Spaltensummen.

U Vektor der Länge M. Während des Programmablaufs Hilfsspeicher. Nach Ende der Rechnung enthält U die Zeilensummen von OPT.

V Vektor der Länge N. Während des Programmablaufs Hilfsspeicher. Nach Ende der Rechnung enthält V die Spaltensummen von OPT.

TOL Kriterium für die Beendigung der Iteration. Muß vom Benutzer gesetzt werden. Die Iteration wird abgebrochen, falls $|XA(IJ)-X(IJ)| \leq$ TOL für alle IJ = 1,..., M*N.

OBJ Funktionswert im optimalen primalen Punkt, d.h.

$$OBJ = \sum_{IJ=1}^{M*N} (OPT(IJ) - A(IJ))^2$$

IT Benötigte Anzahl der Iterationen.

M Anzahl der Zeilen der Ausgangsmatrix.

N Anzahl der Spalten der Ausgangsmatrix.

Benötigte Unterprogramme

DABS IBM-Systemprogramm

Eingabedaten

A, Z, S, TOL, M, N.

Bemerkungen

Das Programm setzt voraus, daß alle auftretenden (M,N)-Matrizen (also insbesondere die Eingabematrix A) zeilenweise in Vektoren der Länge M*N gespeichert sind. Da das Programm als Subroutine geschrieben ist, braucht deshalb für die indizierten Variablen der entsprechende Speichersatz nur im Hauptprogramm, das vom Benutzer erstellt werden muß, reserviert zu werden.

Für das duale Programm aus Kapitel 4 lauten die Kuhn-Tucker-Bedingungen:
Ein zulässiger Punkt λ ist genau dann optimal, wenn ein $\alpha \geq o$ existiert, so daß

$$\frac{1}{2} P\lambda + x_o = \alpha$$

$$\lambda^T \alpha = o.$$

Nach Abbruch der Iteration mit Hilfe der Toleranzgröße TOL enthält der Parameter XA die Werte

XA(IJ) = X(IJ)*OPT(IJ),IJ=1,...,M N ,

wobei X der letzte Iterationspunkt ist. Demgemäß kann der Vektor XA als Test auf das Erfülltsein der Kuhn-Tucker-Bedingungen benutzt werden.

Aus der Beschreibung von MESTP ist ersichtlich, daß das Programm

5*M*N + 2*(M+N)

Speicherplätze für die dimensionierten Variablen benötigt.

Subroutine MESTPV

Der Name dient als Abkürzung für "Minimierung des euklidischen Skalarprodukts über einem Transportpolyeder unter Berücksichtigung von Vorgabewerten."

Das Programm setzt voraus, daß die Eingabematrix A und die neuen Zeilen- bzw. Spaltensummen Z bzw. S in dem vom Benutzer erstellten Hauptprogramm gemäß (25) aus Kapitel 6 modifiziert worden sind. MESTPV liefert dann den optimalen Punkt des Problems (26). Demgemäß erhält man den gesuchten optimalen Punkt, indem wiederum im Hauptprogramm OPT nach Lemma 6.1, 6.2 verändert wird.

Zweck

Lösung des Nichtnegativen Matrixproblems mit Vorgabewerten als Normminimierungsaufgabe bezüglich der euklidischen Norm.

Methode

Lösung des dualen Programms durch zyklischen Koordinatenabstieg.

Aufruf

```
CALL MESTPV(A,R,XO,X,XA,H,OPT,Z,S,U,V,TOL,OBJ,IT,M,N)
```

Beschreibung der Parameter

A	Vektor der Länge M*N. Gemäß (25) aus Kapitel 6 modifizierte Ausgangsmatrix.
R	Vektor der Länge M*N. Die Elemente von R müssen vor Aufruf vom MESTPV gesetzt werden gemäß

$$R(IJ):= \begin{cases} 1, & \text{falls dem Index IJ ein Vorgabewert entspricht, IJ=1,..., M*N} \\ 0 & \text{sonst.} \end{cases}$$

Während des Programmablaufs wird R verändert und dient als Hilfsspeicher.

H Vektor der Länge M*N. Hilfsspeicher.

Z Vektor der Länge M. Gemäß (25) aus Kapitel 6 modifizierte neue Zeilensummen.

S Vektor der Länge N. Gemäß (25) aus Kapitel 6 modifizierte neue Spaltensummen.

XO,X,XA,OPT,U,V,TOL,OBJ,IT,M,N: Analog zu MESTP

Benötigte Unterprogramme

DABS IBM Systemprogramm

SIGMA Siehe anliegende Programmliste.
SIGMA berechnet Zeilen- und Spaltensummen einer vorgegebenen (M,N)-Matrix.

Eingabedaten

A, R, Z, S, TOL, M, N

Bemerkungen

Wie bei MESTP

Der benötigte Speicherplatz für die indizierten Variablen beträgt

$$7*M*N + 2*(M+N).$$

```
0001                SUBROUTINE MESTP(A,XO,X,XA,OPT,Z,S,U,V,TOL,OBJ,IT,M,N)                00000100
0002                IMPLICIT REAL*8 (A-H,O-Z)                                             00000200
0003                DIMENSION A(1),XO(1),X(1),XA(1),OPT(1),Z(1),S(1),U(1),V(1)            00000300
0004                LOGICAL W                                                             00000400
                  C                                                                       00000500
                  C                                                                       00000600
0005                9000 FORMAT(1H1,'EINGABEDATEN FEHLERHAFT'///)                         00000700
0006                9001 FORMAT(1H ,'SUMME DER ZEILENSUMMEN',2X,G16.6)                    00000800
0007                9002 FORMAT(1H ,'SUMME DER SPALTENSUMMEN',2X,G16.6///)                00000900
0008                9003 FORMAT(1H ,'ELEMENT MINIMALER NORM HAT NUR NICHT-NEGATIVE KOMPONEN00001000
                       *TEN'///)                                                          00001100
                  C                                                                       00001200
                  C                                                                       00001300
                  C     KORREKTHEIT DER EINGABEDATEN PRUEFEN                              00001400
                  C                                                                       00001500
                  C                                                                       00001600
0009                SI1=0.D0                                                            00001700
0010                DO 1 I=1,M                                                          00001800
0011                 1  SI1=SI1+Z(I)                                                    00001900
0012                SI2=0.D0                                                            00002000
0013                DO 2 J=1,N                                                          00002100
0014                 2  SI2=SI2+S(J)                                                    00002200
0015                IF(SI1.EQ.SI2) GOTO 10                                              00002300
0016                WRITE (6,9000)                                                      00002400
0017                WRITE (6,9001) SI1                                                  00002500
0018                WRITE (6,9002) SI2                                                  00002600
0019                GOTO 60                                                             00002700
                  C                                                                       00002800
                  C                                                                       00002900
0020                10  MN=M*N                                                          00003000
0021                IT=0                                                                00003100
                  C                                                                       00003200
                  C                                                                       00003300
                  C     XO BERECHNEN                                                      00003400
                  C                                                                       00003500
```

```
      C                                                                 00003600
0022        JSTART=1-N                                                  00003700
0023        JEND=0                                                      00003800
0024        DO 11 I=1,M                                                 00003900
0025        U(I)=0.D0                                                   00004000
0026        JSTART=JSTART+N                                             00004100
0027        JEND=JEND+N                                                 00004200
0028        DO 11 IJ=JSTART,JEND                                        00004300
0029        U(I)=U(I)+A(IJ)                                             00004400
0030     11 CONTINUE                                                    00004500
0031        ISTART=0                                                    00004600
0032        IEND=MN-N                                                   00004700
0033        DO 12 J=1,N                                                 00004800
0034        V(J)=0.D0                                                   00004900
0035        ISTART=ISTART+1                                             00005000
0036        IEND=IEND+1                                                 00005100
0037        DO 12 IJ=ISTART,IEND,N                                      00005200
0038        V(J)=V(J)+A(IJ)                                             00005300
0039     12 CONTINUE                                                    00005400
0040        SI1=0.D0                                                    00005500
0041        DO 13 J=1,N                                                 00005600
0042     13 SI1=SI1+V(J)                                                00005700
0043        SI2=0.D0                                                    00005800
0044        DO 14 I=1,M                                                 00005900
0045     14 SI2=SI2+Z(I)                                                00006000
0046        RMN=1.D0/DFLOAT(MN)                                         00006100
0047        RM=1.D0/DFLOAT(M)                                           00006200
0048        RN=1.D0/DFLOAT(N)                                           00006300
0049        SIG=(SI1-SI2)*RMN                                           00006400
0050        W=.TRUE.                                                    00006500
0051        JSTART=1-N                                                  00006600
0052        JEND=0                                                      00006700
0053        DO 15 I=1,M                                                 00006800
0054        SI=SIG+(Z(I)-U(I))*RN                                       00006900
0055        J=0                                                         00007000
```

```
0056                JSTART=JSTART+N                                          00007100
0057                JEND=JEND+N                                              00007200
0058                DO 15 IJ=JSTART,JEND                                     00007300
0059                J=J+1                                                    00007400
0060                XO(IJ)=A(IJ)+(S(J)-V(J))*RM+SI                           00007500
0061                IF(XO(IJ).LT.0.D0) W=.FALSE.                             00007600
0062             15 CONTINUE                                                 00007700
0063                IF(.NOT.W) GOTO 20                                       00007800
0064                WRITE (6,9003)                                           00007900
0065                OBJ=0.D0                                                 00008000
0066                DO 16 IJ=1,MN                                            00008100
0067                EIJ=XO(IJ)-A(IJ)                                         00008200
0068                OBJ=OBJ+EIJ*EIJ                                          00008300
0069             16 CONTINUE                                                 00008400
0070                DO 17 IJ=1,MN                                            00008500
0071             17 OPT(IJ)=XO(IJ)                                           00008600
0072                GOTO 50                                                  00008700
                 C                                                          00008800
                 C                                                          00008900
                 C     ITERATION VORBEREITEN                                00009000
                 C                                                          00009100
                 C                                                          00009200
0073             20 R=DFLOAT(MN)/DFLOAT((M-1)*(N-1))                        00009300
0074                R=-2.D0*R                                                00009400
0075                DO 21 IJ=1,MN                                            00009500
0076                X(IJ)=0.D0                                               00009600
0077                IF(XO(IJ).LT.0.D0) X(IJ)=R*XO(IJ)                        00009700
0078             21 CONTINUE                                                 00009800
0079                JSTART=1-N                                               00009900
0080                JEND=0                                                   00010000
0081                DO 22 I=1,M                                              00010100
0082                U(I)=0.D0                                                00010200
0083                JSTART=JSTART+N                                          00010300
0084                JEND=JEND+N                                              00010400
0085                DO 22 IJ=JSTART,JEND                                     00010500
```

```
0086            U(I)=U(I)+X(IJ)                                          00010600
0087         22 CONTINUE                                                 00010700
0088            ISTART=0                                                 00010800
0089            IEND=MN-N                                                00010900
0090            DO 23 J=1,N                                              00011000
0091            V(J)=0.D0                                                00011100
0092            ISTART=ISTART+1                                          00011200
0093            IEND=IEND+1                                              00011300
0094            DO 23 IJ=ISTART,IEND,N                                   00011400
0095            V(J)=V(J)+X(IJ)                                          00011500
0096         23 CONTINUE                                                 00011600
0097            SI=0.D0                                                  00011700
0098            DO 24 J=1,N                                              00011800
0099         24 SI=SI+V(J)                                               00011900
0100            R=DFLOAT(MN)/DFLOAT((M-1)*(N-1))                         00012000
0101            DO 25 IJ=1,MN                                            00012100
0102         25 XO(IJ)=2.D0*XO(IJ)                                       00012200
               C                                                         00012300
               C                                                         00012400
               C     ITERATIONSTEIL                                      00012500
               C                                                         00012600
               C                                                         00012700
0103         40 IT=IT+1                                                  00012800
0104            DO 30 IJ=1,MN                                            00012900
0105         30 XA(IJ)=X(IJ)                                             00013000
0106            JSTART=1-N                                               00013100
0107            JEND=0                                                   00013200
0108            DO 31 I=1,M                                              00013300
0109            UI=U(I)                                                  00013400
0110            J=0                                                      00013500
0111            JSTART=JSTART+N                                          00013600
0112            JEND=JEND+N                                              00013700
0113            DO 32 IJ=JSTART,JEND                                     00013800
0114            J=J+1                                                    00013900
0115            XIJ=X(IJ)                                                00014000
```

```
0116              XOIJ=XO(IJ)                                                    00014100
0117              UI=UI-XIJ                                                      00014200
0118              VJ=V(J)-XIJ                                                    00014300
0119              SI=SI-XIJ                                                      00014400
0120              AL=VJ*RM+UI*RN-SI*RMN-XOIJ                                     00014500
0121              IF(AL.LE.0.D0) GOTO 38                                         00014600
0122              XIJ=R*AL                                                       00014700
0123              V(J)=VJ+XIJ                                                    00014800
0124              UI=UI+XIJ                                                      00014900
0125              SI=SI+XIJ                                                      00015000
0126              X(IJ)=XIJ                                                      00015100
0127              GOTO 32                                                        00015200
0128         38   X(IJ)=0.D0                                                     00015300
0129              V(J)=VJ                                                        00015400
0130         32   CONTINUE                                                       00015500
0131              U(I)=UI                                                        00015600
0132         31   CONTINUE                                                       00015700
0133              DO 33 IJ=1,MN                                                  00015800
0134              IF(DABS(XA(IJ)-X(IJ)).GT.TOL) GOTO 40                          00015900
0135         33   CONTINUE                                                       00016000
0136              SI=RMN*SI                                                      00016100
0137              JSTART=1-N                                                     00016200
0138              JEND=0                                                         00016300
0139              DO 34 I=1,M                                                    00016400
0140              UI=RN*U(I)                                                     00016500
0141              SIG=SI-UI                                                      00016600
0142              J=0                                                            00016700
0143              JSTART=JSTART+N                                                00016800
0144              JEND=JEND+N                                                    00016900
0145              DO 34 IJ=JSTART,JEND                                           00017000
0146              J=J+1                                                          00017100
0147              OPT(IJ)=5.0D-1*(XO(IJ)+X(IJ)-RM*V(J)+SIG)                      00017200
0148              XO(IJ)=5.0D-1*XO(IJ)                                           00017300
0149              XA(IJ)=X(IJ)*OPT(IJ)                                           00017400
0150         34   CONTINUE                                                       00017500
```

```
0151              OBJ=0.DO                                              00017600
0152              DO 35 IJ=1,MN                                         00017700
0153              EIJ=OPT(IJ)-A(IJ)                                     00017800
0154              OBJ=OBJ+EIJ*EIJ                                       00017900
0155         35   CONTINUE                                              00018000
0156         50   JSTART=1-N                                            00018100
0157              JEND=0                                                00018200
0158              DO 36 I=1,M                                           00018300
0159              U(I)=0.DO                                             00018400
0160              JSTART=JSTART+N                                       00018500
0161              JEND=JEND+N                                           00018600
0162              DO 36 IJ=JSTART,JEND                                  00018700
0163              U(I)=U(I)+OPT(IJ)                                     00018800
0164         36   CONTINUE                                              00018900
0165              ISTART=0                                              00019000
0166              IEND=MN-N                                             00019100
0167              DO 37 J=1,N                                           00019200
0168              V(J)=0.DO                                             00019300
0169              ISTART=ISTART+1                                       00019400
0170              IEND=IEND+1                                           00019500
0171              DO 37 IJ=ISTART,IEND,N                                00019600
0172              V(J)=V(J)+OPT(IJ)                                     00019700
0173         37   CONTINUE                                              00019800
0174         60   RETURN                                                00019900
0175              END                                                   00020000
```

```
0001                  SUBROUTINE MESTPV(A,R,XO,X,XA,H,OPT,Z,S,U,V,TOL,OBJ,IT,M,N)        00000010
0002                  IMPLICIT REAL*8 (A-H,O-Z)                                          00000020
0003                  DIMENSION A(1),R(1),XO(1),X(1),XA(1),H(1),OPT(1),                  00000030
                     *          Z(1),S(1),U(1),V(1)                                    00000040
                                                                                       00000050
0004                  LOGICAL W                                                        00000060
                C                                                                      00000070
                C                                                                      00000080
0005                 9000 FORMAT(1H1,'DIE ZU DEN VORGABEWERTEN GEHOERENGE ZEILE DER NEBENBED00000090
                     *INGUNGSMATRIX IST LINEAR ABHAENGIG VON DEN ZEILEN DER TRANSPORTMAT00000100
                     *RIX.DESHALB ABBRUCH')                                            00000110
0006                 9001 FORMAT(1H1,'ELEMENT MINIMALER NORM HAT NUR NICHT-NEGATIVE KOMPONEN00000120
                     *TEN'///)                                                         00000130
                C                                                                      00000140
                C                                                                      00000150
0007                  MN=M*N                                                           00000160
0008                  RMN=1.DO/DFLOAT(MN)                                              00000170
0009                  RM=1.DO/DFLOAT(M)                                                00000180
0010                  RN=1.DO/DFLOAT(N)                                                00000190
0011                  IT=0                                                             00000200
                C                                                                      00000210
                C                                                                      00000220
                C     MINIMUM OHNE VORZEICHENBESCHRAENKUNG BERECHNEN                   00000230
                C                                                                      00000240
                C                                                                      00000250
0012                  CALL SIGMA(A,U,V,M,N,SI1)                                        00000260
0013                  SI2=0.DO                                                         00000270
0014                  DO 1 I=1,M                                                       00000280
0015                1 SI2=SI2+Z(I)                                                     00000290
0016                  SIG=(SI1-SI2)*RMN                                                00000300
0017                  JSTART=1-N                                                       00000310
0018                  JEND=0                                                           00000320
0019                  DO 2 I=1,M                                                       00000330
0020                  SI=SIG+(Z(I)-U(I))*RN                                            00000340
0021                  J=0
```

```
0022              JSTART=JSTART+N                                        00000350
0023              JEND=JEND+N                                            00000360
0024              DO 2 IJ=JSTART,JEND                                    00000370
0025              J=J+1                                                  00000380
0026              XO(IJ)=A(IJ)+(S(J)-V(J))*RM+SI                         00000390
0027          2   CONTINUE                                               00000400
0028              CALL SIGMA(R,U,V,M,N,SIG)                              00000410
0029              FN=DFLOAT(N)                                           00000420
0030              FM=DFLOAT(M)                                           00000430
0031              FMN=DFLOAT(MN)                                         00000440
0032              NZERO=0                                                00000450
0033              JSTART=1-N                                             00000460
0034              JEND=0                                                 00000470
0035              DO 3 I=1,M                                             00000480
0036              SI=SIG-FM*U(I)                                         00000490
0037              J=0                                                    00000500
0038              JSTART=JSTART+N                                        00000510
0039              JEND=JEND+N                                            00000520
0040              DO 3 IJ=JSTART,JEND                                    00000530
0041              J=J+1                                                  00000540
0042              H(IJ)=FMN*R(IJ)-FN*V(J)+SI                             00000550
0043              IF(H(IJ).EQ.0.D0) NZERO=NZERO+1                        00000560
0044          3   CONTINUE                                               00000570
0045              IF(NZERO.LT.MN) GOTO 4                                 00000580
0046              WRITE (6,9000)                                         00000590
0047              GOTO 60                                                00000600
0048          4   DO 5 IJ=1,MN                                           00000610
0049          5   H(IJ)=RMN*H(IJ)                                        00000620
0050              PROD=0.D0                                              00000630
0051              DO 6 IJ=1,MN                                           00000640
0052              HIJ=H(IJ)                                              00000650
0053          6   PROD=PROD+HIJ*HIJ                                      00000660
0054              RPROD=1.D0/PROD                                        00000670
0055              RXO=0.D0                                               00000680
```

```
0056                DO 7 IJ=1,MN                                          00000690
0057                RXO=RXO+R(IJ)*XO(IJ)                                  00000700
0058                H(IJ)=RPROD*H(IJ)                                     00000710
0059           7    CONTINUE                                              00000720
0060                W=.TRUE.                                              00000730
0061                DO 8 IJ=1,MN                                          00000740
0062                XO(IJ)=XO(IJ)-H(IJ)*RXO                               00000750
0063                IF(XO(IJ).LT.0.DO) W=.FALSE.                          00000760
0064           8    CONTINUE                                              00000770
0065                IF(.NOT.W) GOTO 20                                    00000780
0066                WRITE (6,9001)                                        00000790
0067                OBJ=0.DO                                              00000800
0068                DO 9 IJ=1,MN                                          00000810
0069                EIJ=XO(IJ)-A(IJ)                                      00000820
0070                OBJ=OBJ+EIJ*EIJ                                       00000830
0071           9    CONTINUE                                              00000840
0072                DO 10 IJ=1,MN                                         00000850
0073           10   OPT(IJ)=XO(IJ)                                        00000860
0074                GOTO 50                                               00000870
                C                                                         00000880
                C                                                         00000890
                C     ITERATION VORBEREITEN                               00000900
                C                                                         00000910
                C                                                         00000920
0075           20   RR=(DFLOAT(M-1)*DFLOAT(N-1))/DFLOAT(MN)               00000930
0076                WPROD=DSQRT(PROD)                                     00000940
0077                DO 21 IJ=1,MN                                         00000950
0078                HIJ=WPROD*H(IJ)                                       00000960
0079                R(IJ)=1.DO/(RR-HIJ*HIJ)                               00000970
0080                H(IJ)=HIJ                                             00000980
0081                XA(IJ)=-2.DO*R(IJ)                                    00000990
0082           21   CONTINUE                                              00001000
0083                DO 22 IJ=1,MN                                         00001010
0084                X(IJ)=0.DO                                            00001020
```

```
0085              IF(XO(IJ).LT.0.DO) X(IJ)=XA(IJ)*XO(IJ)                    00001030
0086           22 CONTINUE                                                  00001040
0087              CALL SIGMA(X,U,V,M,N,SI)                                  00001050
0088              HX=0.DO                                                   00001060
0089              DO 23 IJ=1,MN                                             00001070
0090              HX=HX+H(IJ)*X(IJ)                                         00001080
0091              XO(IJ)=2.DO*XO(IJ)                                        00001090
0092           23 CONTINUE                                                  00001100
                C                                                          00001110
                C                                                          00001120
                C     ITERATIONSTEIL                                       00001130
                C                                                          00001140
                C                                                          00001150
0093           40 IT=IT+1                                                  00001160
0094              DO 30 IJ=1,MN                                             00001170
0095           30 XA(IJ)=X(IJ)                                             00001180
0096              JSTART=1-N                                                00001190
0097              JEND=0                                                    00001200
0098              DO 31 I=1,M                                               00001210
0099              UI=U(I)                                                   00001220
0100              J=0                                                       00001230
0101              JSTART=JSTART+N                                           00001240
0102              JEND=JEND+N                                               00001250
0103              DO 32 IJ=JSTART,JEND                                      00001260
0104              J=J+1                                                     00001270
0105              XIJ=X(IJ)                                                 00001280
0106              XOIJ=XO(IJ)                                               00001290
0107              HIJ=H(IJ)                                                 00001300
0108              RIJ=R(IJ)                                                 00001310
0109              UI=UI-XIJ                                                 00001320
0110              VJ=V(J)-XIJ                                               00001330
0111              SI=SI-XIJ                                                 00001340
0112              HX=HX-HIJ*XIJ                                             00001350
0113              AL=VJ*RM+UI*RN+HIJ*HX-SI*RMN-XOIJ                         00001360
```

```
0114              IF(AL.LE.0.D0) GOTO 38                                        00001370
0115              XIJ=RIJ*AL                                                    00001380
0116              V(J)=VJ+XIJ                                                   00001390
0117              UI=UI+XIJ                                                     00001400
0118              SI=SI+XIJ                                                     00001410
0119              HX=HX+HIJ*XIJ                                                 00001420
0120              X(IJ)=XIJ                                                     00001430
0121              GOTO 32                                                       00001440
0122         38   X(IJ)=0.D0                                                    00001450
0123              V(J)=VJ                                                       00001460
0124         32   CONTINUE                                                      00001470
0125              U(I)=UI                                                       00001480
0126         31   CONTINUE                                                      00001490
0127              DO 33 IJ=1,MN                                                 00001500
0128              IF(DABS(XA(IJ)-X(IJ)).GT.TOL) GOTO 40                         00001510
0129         33   CONTINUE                                                      00001520
0130              SI=RMN*SI                                                     00001530
0131              JSTART=1-N                                                    00001540
0132              JEND=0                                                        00001550
0133              DO 34 I= 1,M                                                  00001560
0134              UI=RN*U(II                                                    00001570
0135              SIG=SI-UI                                                     00001580
0136              J=0                                                           00001590
0137              JSTART=JSTART+N                                               00001600
0138              JEND=JEND+N                                                   00001610
0139              DO 34 IJ=JSTART,JEND                                          00001620
0140              J=J+1                                                         00001630
0141              OPT(IJ)=5.0D-1*(XO(IJ)+X(IJ)-RM*V(J)+SIG-H(IJ)*HX)            00001640
0142              XO(IJ)=5.0D-1*XO(IJ)                                          00001650
0143              XA(IJ)=X(IJ)*OPT(IJ)                                          00001660
0144         34   CONTINUE                                                      00001670
0145              OBJ=0.D0                                                      00001680
0146              DO 35 IJ=1,MN                                                 00001690
0147              EIJ=OPT(IJ)-A(IJ)                                             00001700
```

```
0148              OBJ=OBJ+EIJ*EIJ                                        00001710
0149          35  CONTINUE                                               00001720
0150          50  CALL SIGMA(OPT,U,V,M,N,SI)                             00001730
0151          60  RETURN                                                 00001740
0152              END                                                    00001750
```

```
0001              SUBROUTINE SIGMA(X,U,V,M,N,S)                          00001760
0002              DIMENSION X(1),U(1),V(1)                               00001770
0003              DOUBLE PRECISION X,U,V,S                               00001780
                C                                                        00001790
0004              MN=M*N                                                 00001800
0005              JSTART=1-N                                             00001810
0006              JEND=0                                                 00001820
0007              DO 1 I=1,M                                             00001830
0008              U(I)=0.D0                                              00001840
0009              JSTART=JSTART+N                                        00001850
0010              JEND=JEND+N                                            00001860
0011              DO 1 IJ=JSTART,JEND                                    00001870
0012              U(I)=U(I)+X(IJ)                                        00001880
0013            1 CONTINUE                                               00001890
0014              ISTART=0                                               00001900
0015              IEND=MN-N                                              00001910
0016              DO 2 J=1,N                                             00001920
0017              V(J)=0.D0                                              00001930
0018              ISTART=ISTART+1                                        00001940
0019              IEND=IEND+1                                            00001950
0020              DO 2 IJ=ISTART,IEND,N                                  00001960
0021              V(J)=V(J)+X(IJ)                                        00001970
0022            2 CONTINUE                                               00001980
0023              S=0.D0                                                 00001990
0024              DO 3 J=1,N                                             00002000
0025            3 S=S+V(J)                                               00002010
0026              RETURN                                                 00002020
0027              END                                                    00002030
```

Literaturangaben

[1] J. Aczel, On different characterizations of entropies, in: M. Behara, K. Krickeberg, J. Wolfowitz (eds), Probability and Information Theory, Lecture Notes in Mathematics, Nr. 89, Springer, 1969

[2] A. Albert, Regression and the Moore-Penrose pseudo-inverse, Academic Press, 1972

[3] R. Ash, Information Theory, Wiley-Interscience, New York, 1967

[4] M. Bacharach, Biproportional matrices and input-output change, Cambridge University Press, 1970

[5] A. Ben-Israel, T.N.E. Greville, Generalized inverses, Wiley-Interscience, New York, 1974

[6] T.L. Boullion, P. Odell, Generalized inverse matrices, Wiley-Interscience, New York, 1971

[7] R.E. Cline, Representations for the generalized inverse of a partitioned matrix, SIAM J. Appl. Math. 12 (1964), 588-600

[8] R.E. Cline, Note on the generalized inverse of the product of matrices, SIAM Rev. 6 (1964), 57-58

[9] W.E. Deming, F.F. Stephan, On the least squares adjustment of a sampled frequency table when the expected marginal totals are known, Ann. Math. Statistics 11 (1940), 427-444

[10] D. Friedlander, A technique for estimating a contingency table, given the marginal totals and some supplementary data, J. Royal Stat. Soc., Series A 124 (1961), 412-420

[11] T.N.E. Greville, Some applications of the pseudoinverse of a matrix, SIAM Rev. 2 (1960), 15-22

[12] R.B. Holmes, A course on optimization and approximation, Lecture Notes in Mathematics Nr. 257, Springer, New York, 1972

[13] H.J. Jaksch, K. Conrad, Zur Schätzung von Input-Output-Tabellen aus ihren Reihensummen, Jahrb. f. Nationalökonomie und Statistik 186 (1971), 131-138

[14] V. Klee, C. Witzgall, Facets and vertices of transportation polytopes, in: Dantzig, Veinott (eds.), Mathematics of the decision sciences Part I, 257-282

[15] A. Land, S. Powell, Fortran Codes for mathematical programming, Wiley-Interscience, New York, 1973

[16] T.I. Matuszewski, P.R. Pitts, J.A. Sawyer, Linear programming estimates of changes in input coefficients, Canad. J. Econ. Pol. Sciences 30 (1964) 203-210

[17] D.L. Nelson, T.O. Lewis, T.L. Boullion, A quadratic programming technique using matrix pseudoinverses, Indust. Math. 21 (1971), 1-21

[18] R. Penrose, A generalized inverse for matrices, Proc. Cambridge Philos. Soc. 51 (1955), 406-413

[19] L.D. Pyle, The generalized inverse in linear programming - basic structure, SIAM J. Appl. Math. 22 (1972), 335-355

[20] L.D. Pyle, R.E. Cline, The generalized inverse in linear programming. - interior gradient projection methods, SIAM J. Appl. Math. 24 (1973), 511-534

[21] D.G. Shankland, Quadratic programming using generalized inverses; Technical report 75-2, 1972, Air Force Institute of Technology

[22] C.E. Shannon, A mathematical theory of communication, Bell System Techn. J. 27 (1948), 379-423, 623-656

[23] J. Schintke, Modell der doppelten Proportionalität zur Schätzung von nichtnegativen Matrizen, insbesondere Input-Output-Tabellen, Angew. Inform. 4 (1973), 153-156

[24] E. Schultze, Einführung in die mathematischen Grundlagen der Informationstheorie, Lecture Notes in Operations Research and Mathematical Economics Nr. 9, Springer, New York, 1969

[25] R. Stäglin, Aufstellung von Input-Output-Tabellen, DIW-Beiträge zur Konjunkturforschung, Heft 4, 1968

[26] R. Stäglin, Modop- ein Verfahren zur Erstellung empirischer Transaktionsmatrizen, Berichte aus dem Institut für Statistik und Versicherungsmathematik und aus dem Institut für Angewandte Statistik der Freien Universität Berlin, Heft 15, 69-81

[27] R. Stone, J. Bates, M. Bacharach, Input-Output relationships 1954-1966, A programme for growth, paper no. 3, University of Cambridge, London, 1963

[28] H. Theil, Economics and information theory, North Holland, Amsterdam, 1967

[29] F. Topsóe, Informationstheorie, Teubner Studienbücher Mathematik, Stuttgart, 1974

[30] W. Vogel, Lineares Optimieren, Akademische Verlagsgesellschaft, Leipzig, 1967

Hiermit versichere ich an Eides Statt, daß ich die vorliegende Dissertation ohne unerlaubte fremde Hilfe angefertigt und andere als die in der Dissertation angegebenen Hilfsmittel nicht benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Lebenslauf

30. 7. 1944	geboren in Dessau als Sohn des Dipl.-Ing. Hans-Georg Braun und seiner Ehefrau Käte, geb. Lehmann. Im selben Jahr Übersiedlung nach Bernburg/Saale.
1950	Umzug nach Berlin (Ost).
1951-1956	Besuch der Volksschule in Berlin-Pankow, nach der Flucht in die Bundesrepublik am 8. 2. 1956 zunächst Besuch der Schule im Flüchtlingslager Wentorf bei Hamburg, anschließend des Hansa-Gymnasiums, Hamburg-Bergedorf.
1957-1965	Nach der Übersiedlung nach Bonn Besuch des Staatlichen-Nicolaus-Cusanus-Gymnasiums, Bonn-Bad Godesberg, Abitur.
1965-1967	Militärdienst
1967-1971	Studium der Mathematik an der Rheinischen Friedrich-Wilhelms-Universität, Bonn, Abschluß des Studiums mit dem Diplom in Mathematik.
1971-1972	Wissenschaftliche Hilfskraft am Mathematischen Institut der Universität Bonn.
seit 1972	Tätigkeit am Institut für Ökonometrie und Operations Research, zunächst als Verwalter der Stelle eines Wissenschaftlichen Assistenten, seit dem 22. 5. 1973 als Wissenschaftlicher Assistent.